IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 11, 2020, accepted October 28, 2020, date of publication November 3, 2020, date of current version November 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035682

The Vehicle Scheduling Problem of Third-Party Passenger Finished Vehicle Logistics Transportation: Formulation, Algorithms, and Instances

ZHIYUAN SU AND WANTAO LI

School of Modern Post (School of Automation), Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Zhiyuan Su (szyyyp@163.com)

This work was supported in part by the 2015 China Special Fund for Grain-Scientific Research in Public Interest under Grant 201513002.

ABSTRACT In this article, a vehicle scheduling problem of third-party passenger finished vehicle logistics transportation networks is studied. An integer programming model of open heterogeneous fleet pickup and delivery problem with time windows and split load (OHFPDPTWSL) is established to maximize the total profit, and a hybrid parallel heuristic algorithm combining with path buffer clustering operator (PBC), multi-mark split operator (MMS) and four variable neighborhood search operators (PBCMMSVNSHPA) is proposed to solve this problem with high quality in a relatively short time. The PBC operator with three different clustering types can effectively cluster the orders and the transport vehicles before the route planning, and the MMS operator can greatly reduce the complexity and computation of the path planning at the expense of little algorithm precision. Then a set of instances which represents the realistic characters of OHFPDPTWSL modified from benchmark instances is introduced. The experimental results on these instances show that PBCMMSVNSHPA is suitable for real-time requirement or large-scale dataset, and the experimental results on the actual instance of an enterprise show that this algorithm can solve the instance with 200 orders and 500 vehicles within 3 minutes.

INDEX TERMS Passenger finished vehicle logistics, open heterogeneous fleet, pickup and delivery problem with time windows and split load, path buffer clustering, multi-mark split.

I. INTRODUCTION

In recent years, with the rapid growth of China's automobile production and sales, the automotive logistics industry has achieved rapid development. In 2018, the passenger finished vehicle logistics market in China exceeded 800 billion yuan, of which transportation cost accounted for more than 40%. The main mode of domestic passenger finished vehicle transportation is road transportation of which the proportion is more than 85%, but it still exists a problem that the empty driving rate of vehicles is as high as 40% [1].

The passenger finished vehicle logistics (PFVL) refers to the logistics process that the passenger vehicles are finally handed over to the customer at the designated time and place after having been produced from the host factory, through storage, transportation, and information transmission [2]. PFVL is located downstream of the automobile industry chain, which is the bridge between production and sales. As shown in Fig. 1, the domestic PFVL industry generally follows the "two-level distribution" mode at present: The passenger finished vehicles enter the manufacturer's vehicle distribution center (VDC) after being produced from the host factory. The transportation from the vehicle distribution center (VDC) to the vehicle storage center (VSC) is the first transportation, and this stage of transportation is usually completed by the manufacturer itself. The transportation from the vehicle storage center (VSC) to the 4S automobile sale center (ASC) is the second transportation and generally completed by third-party logistics (TPL) [2].

The third-party logistics transportation network of passenger finished vehicles mainly includes vehicle storage centers (VSC), transport vehicles of TPL enterprises, and 4S

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

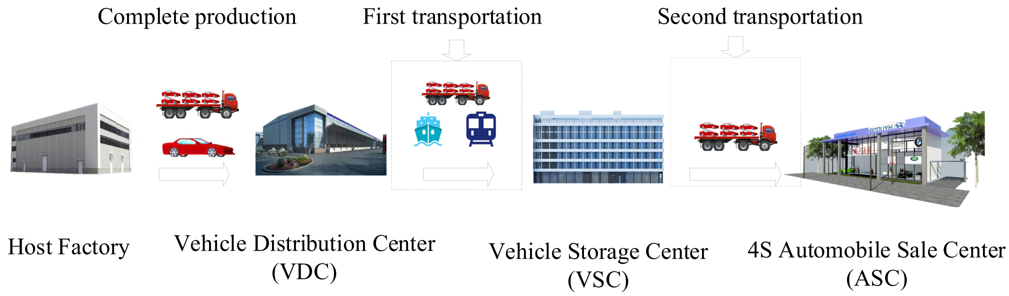

FIGURE 1. Transportation mode of passenger vehicle logistics.

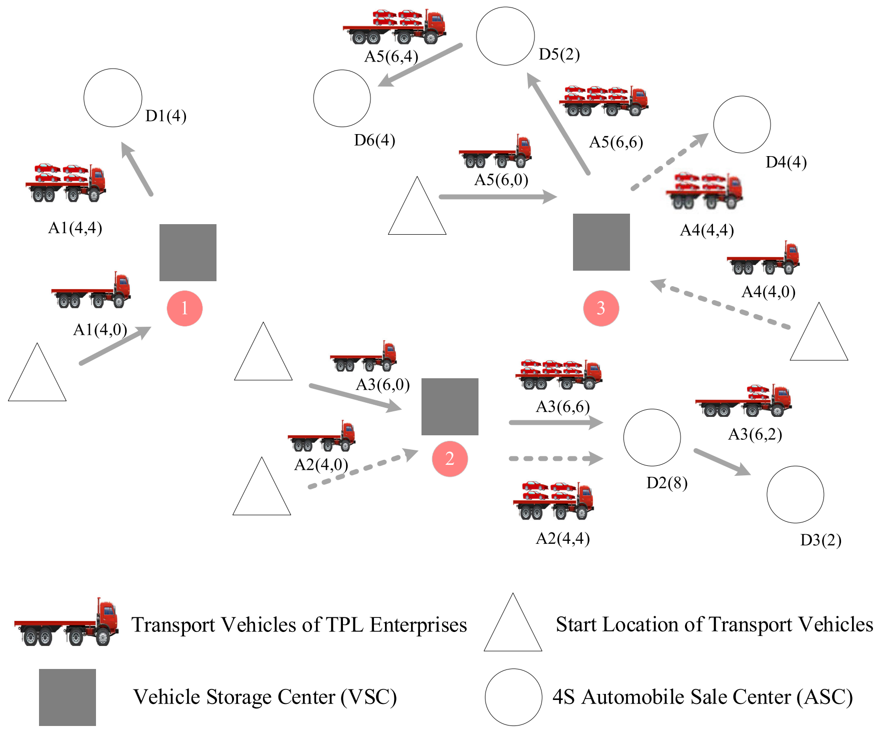

FIGURE 2. Schematic diagram of passenger finished vehicle logistics transportation network.

automobile sale centers (ASC). The transport vehicles start from a random location (generally these vehicles stop nearest to the VSC) and transport the passenger finished vehicle from the VSC to the designated 4S automobile sales center. As shown in Fig. 2, there are 5 transport vehicles, 3 vehicle storage centers (VSC), and 6 4S automobile sale centers (ASC). The symbol below the icon of the transport vehicle represents the vehicle no., and the number in brackets represents its rated capacity and quantity of passenger vehicles currently transported respectively. Equally the symbol below the icon of the SC represents the ASC no., and the number in brackets represents the request of the ASC. From the figure, we can find that the requests of D2 ASC is split and transported by A2 and A3 transport vehicle from No.2 VSC together. The logistics transportation network of passenger finished vehicles is quite different from the traditional transportation network:

1) The requests of third-party passenger finished vehicle logistics have Characteristics of a long-time window, wide geographical distribution.

2) Passenger finished vehicles generally need to use a fleet of special passenger vehicle transport vehicles with different capacities to complete the transportation according to the size and quantity of passenger finished vehicles.

3) To get the maximum total profit, some requests will be split, and some requests will be served in the next scheduling while not all requests must be served in one schedule.

4) The passenger finished vehicle transport vehicle starts from a place nearest the VSC, and usually stops at the place nearest the VSC of the task destination to wait for the next

task after completing a transportation task, while does not return to the original starting position.

Thus, the scheduling problem in the passenger finished vehicle logistics transportation network can be formulated as an open heterogeneous fleet pickup and delivery problem with time windows and split load (OHFPDPTWSL).

The main contributions of this article are as follows:

1) The integer programming model of the vehicle scheduling problem in passenger finished vehicle logistics transportation network is established to maximize the total profit. In this model, there is no distribution center in the transportation network and the heterogeneous fleet of transport vehicles starts from a random location.

2) A path buffer clustering algorithm based on the actual graphical route path buffer including three different types of buffer clustering operators is proposed to cluster the request before the transport vehicle route planning. And it is more suitable for practical engineering applications.

3) Multi-mark split algorithm proposed in this pager can greatly reduce the complexity and computation of the path planning algorithm, and can help to solve the problem with high quality in a relatively short time at the expense of little algorithm precision.

4) A new benchmark which represents the realistic characters of OHFPDPTWSL modified from the instance of Li and Lim [33] by removing the distribution center and adding vehicles with a certain proportion is provided.

The rest of the paper is organized as follows:

1) A brief literature review on Passenger Finished Vehicle Logistics and Split Delivery Vehicle Routing Problem is provided in Section 2.
2) A maximum profit mathematical formulation of this problem is provided in Section 3.
3) A parallel hybrid heuristic algorithm is developed in Section 4.
4) The algorithm is validated on the extended data set of Solomon and a real-world case in Section 5.
5) In Section 6, the conclusion and potential future extensions are discussed.

II. LITERATURE REVIEW

OHFPDPTWSL is an NP-hard problem, which is more difficult for its split load and pickup and delivery constraints [3]. As far as we know, there are few researches completely consistent with OHFPDPTWSL. The related researches are Passenger Finished Vehicle Logistics (PFVL) and Split Delivery Vehicle Routing Problem (SDVRP).

A. PASSENGER FINISHED VEHICLE LOGISTICS

At present, many vehicle logistics companies still rely on manual operation when facing complex transportation tasks, with low efficiency and high cost [4]. There are few researches on the passenger finished vehicle logistics routing optimization. Xue-Ting *et al.* [5] aimed at the logistics planning problem of different specifications of transport vehicles for different specifications of passenger vehicles, proposed the two-stage greedy algorithm for redundant vehicles based on integer programming. It can provide a good loading scheme for the finished vehicle logistics problem under different complexity. The example results show that the utilization rate of transport vehicles can reach more than 90%, but their research focuses on vehicle allocation only. Hu *et al.* [6] established the mixed-integer integer linear programming (MILP) model of the finished vehicle transporter routing problem (FVTRP) considering the loading mode. They used the heuristic method to solve the model, and compared with the results of some commercial solver, the algorithm was more effective in solving the medium-sized problem, but their research does not consider the split load. Jiang *et al.* [7] put forward the vehicle logistics routing optimization network under the two resource sharing modes of shared vehicle and shared vehicle distribution center, constructed the mathematical optimization model aiming at the minimum total vehicle transportation cost, and designed a heuristic algorithm based on genetic algorithm to solve the problem. The experimental results show that the resource sharing model can reduce the total cost of enterprises by more than 30%.

These researches all do not focus on the max profit of the enterprise by completing the transportation in the passenger finished vehicle logistics practical in which the customer's request can be served in a relativity long time window and can be split into multiple segments.

B. SPLIT DELIVERY VEHICLE ROUTING PROBLEM

SDVRP was first proposed by Dror and Trudeau [8], who proved that SDVRP has advantages in both transportation distance and vehicle quantity. Since Dror and Trudeau [8] first proposed SDVRP, many scholars have studied the solution difficulty, the characteristics of the solution, the upper and lower bounds of the solution, and the solution algorithm. The characteristic of the optimal solution of SDVRP is the key to study the problems in this field and was studied in earlier research. Dror and Trudeau [8] first demonstrated and proposed the definition of k-split cycle and two basic characteristics of the optimal solution of SDVRP. Based on this important conclusion, Archetti *et al.* [9], [10] and Desaulniers [11] further obtained the supplementary characteristics of the optimal solution of SDVRP. In actual SDVRP, the total demand of a single customer may be greater than the loading capacity of a transport vehicle, and the vehicle loading and customer demand in practical applications are positive integers. Archetti *et al.* [12] and Archetti and Speranza [13] studied this kind of SDVRP, gave the concept of Q-SDVRP and its definition, and proposed the concept of reducibility of SDVRP. The research results show that when the SDVRP distance matrix with an integer demand satisfies the triangle inequality, it can only be reduced if the vehicle capacity is $Q = 2$. The researches on this part are summarized in Table 1. However, since 2011, we have not found any other relevant research in this field.

TABLE 1. Characteristics of SDVRP optimal solution.

Year	Researcher	Characteristics
1989	Dror *et al.* [8]	There is at most one common point for any two paths in the SDVRP optimal solution
1989	Dror *et al.* [8]	k-split cycle does not exist in SDVRP optimal solution
2006	Archetti *et al.* [10]	The number of demands split points in the SDVRP optimal solution is less than the number of paths
2010	Desaulniers *et al.* [11]	Any arc in SDVRP optimal solution can only be accessed at most once
2011	Archetti *et al.* [12]	The number of vehicles in the SDVRP-UF optimal solution is not more than twice the minimum number of vehicles

Archetti *et al.* [12] proved that the relationship between the average customer demand and vehicle loading capacity is the biggest factor affecting the cost savings of SDVRP. Only when this influencing factor meets certain conditions can SDVRP achieve significant cost savings. At the same time, they proved that the cost savings brought by SDVRP are first of all the reduction in the number of vehicles used due to the detachable demand, which is not affected by the distribution of customer points but is affected by the variance of customer demand. Early scholars only considered the case where the demand at each customer point was less than or equal to the vehicle loading capacity. Dror and Trudeau [8] proved by examples that when the difference between customer demand and vehicle loading capacity is small, the solution of SDVRP is not much improved compared with VRP; when the average customer demand is greater than 10% of vehicle loading capacity, demand splitting will bring significant cost savings. Belenguer *et al.* [14] pointed out that when the demand of each customer point is less than the vehicle loading capacity and the actual transportation volume is an integer, SDVRP has a strict lower bound. Archetti *et al.* [15] studied the situation that the demand of customer point is greater than the loading capacity of the vehicle, and analyzed the cost savings brought by SDVRP to VRP from two aspects of vehicle number and driving distance. Relevant research results are shown in Table 2.

The algorithm of SDVRP can be divided into two parts: the exact algorithm and heuristic algorithm. Dror and Trudeau [8] gave the mixed integer programming model of SDVRP-UF and the branch and bound method of its solution, and put forward several effective inequalities according to the characteristics of the optimal solution of the problem. Archetti *et al.* [9] first solved the SDVRP-LF and SDVRP-UF using the branch-price-cut method. They give a heuristic algorithm that uses the solution (column) obtained by the subproblem as the initial solution to find the upper bound of the problem solution. Jin *et al.* [16] proposed a two-stage method for solving SDVRP with effective inequality constraints, including a client point clustering stage and a TSP path arrangement stage. The former aims to determine the set of customer points visited by each vehicle without considering the driving cost of the vehicle, and the latter aims to solve the requested path by TSP. The resulting cost is used as the lower bound of the problem to insert the next iterative clustering process with constraints. Moreno *et al.* [17] proposed the column generation and facet method for solving SDVRP, and the lower bound of the problem solution can be effectively obtained by using this method. They use dynamic programming methods to solve the pricing problem and use two heuristic algorithms to solve the pricing problem to speed up the process. Archetti *et al.* [18] proposed two branch-cut algorithms for SDVRP based on a relaxation model that guarantees the optimal solution to the problem. The first relaxation model is an improved model of Belenguer *et al.* [14], and the second is a newly proposed relaxation model based on vehicle loading. By comparing the two branch-cut methods, the performance of the first improved relaxation model is better.

Heuristic algorithms are often used to solve large-scale SDVRP. Heuristic algorithms applied in existing research literature include classic heuristics, hybrid heuristics, and metaheuristics. Among them, metaheuristics and hybrid heuristics have become the main algorithms for solving SDVRP. Dror and Trudeau [8] designed a two-stage method for solving SDVRP based on neighborhood search and expounded and proved improved heuristic methods k-Split Interchange and

TABLE 2. SDVRP cost savings study (Q is the maximum capacity of the vehicle, d_i is the demand of customer i, Z is the total path length, K is the number of vehicles used).

Time	Researcher	Condition	Conclusion
1989	Droret al. [8]	$d_i \le Q/10$	No obvious improvement, should be solved by VRP
		$Q/10 \le d_i \le Q$	Significant cost savings
2000	Belenguer *et al.* [14]	$d_i \to +\infty$	$\frac{z(VRP)}{z(SDVRP)} \to 2$
2006	Archetti *et al.* [9]	$d_i \ge Q$	$\frac{z(VRP)}{z(SDVRP\text{-UF})} \to 2$
			$\frac{K(VRP)}{K(SDVRP-LF)} \to 2$
2008	Archetti *et al.* [15]	$\frac{Q}{2} \le \frac{1}{N}\sum_{i=1}^{N} d_i \le \frac{3Q}{4}$	Get the most cost savings

Route Addition. This document is the beginning of SDVRP research and has been widely cited and used by subsequent scholars. Belenguer *et al.* [14] introduced the concept of a polyhedron and used the tangent method to solve the SDVRP of an integer programming model based on arc flow. Campos *et al.* [19] proposed a scanning algorithm and used it to solve SDVRP. Yan *et al.* [20] proposed a two-stage method to solve the split demand vehicle routing and scheduling problems with time windows.

At present, many hybrid heuristic algorithms have been applied to solve SDVRP. It turns out that using a hybrid heuristic algorithm which based on an exact algorithm to obtain a better-quality solution than a single heuristic algorithm. Chen *et al.* [21] first proposed a hybrid heuristic algorithm. The initial solution is given by the C-W saving algorithm for solving VRP. An Endpoint Mixed Integer Program (EMIP) model is used to optimally redistribute the endpoints of each line of the current solution. At the same time, they proposed a Variable Length Record-To-Record Travel Algorithm (VRTR) to continuously improve the solution of EMIS. The algorithm proposed by Archetti *et al.* [22] combines the tabu search algorithm and optimization idea of Archetti *et al.* [23]. Firstly, the tabu search algorithm is used to determine the solution space that is most likely to contain high-quality solutions, and then the integer programming model is used to expand the obtained solution space. Jin *et al.* [24] proposed a hybrid heuristic algorithm based on column generation. The column generation method can be used to solve the SDVRP with large customer demand. The columns generated by the algorithm in the problem contain both path and actual distribution quantity information. The price subproblem is solved by the limited search with a bound (LSWB) algorithm. Khmelev and Kochetov [25] proposed variable neighborhood descent (VND) to solve the SDVRP, which is divided into two subproblems: finding the best arrangement and finding the best route of any arrangement. Firstly, the first subproblem is solved by variable neighborhood descent and random tabu search, and then the second algorithm, the other is based on the subproblem is solved by two fast decoding heuristics.

In 2006, Archetti *et al.* [23] used a tabu search algorithm to solve SDVRP, which is the first time the meta-heuristic algorithm was used to solve such problems. Derigs *et al.* [26] relationship proposed a meta-heuristic algorithm based on neighborhood search. Through different neighborhood operations, they get several different meta-heuristic algorithms: simulated annealing, threshold acceptance, memory update, mountain climbing, and location search. Experiments show that mountain climbing has the best performance. Aleman *et al.* [27] and Aleman and Hill [28] proposed two new metaheuristic algorithms. Aleman *et al.* [27] proposed an adaptive memory algorithm, which obtained the initial solution of the problem by a constructing algorithm, and improved it by using the VNS. Aleman and Hill [28] proposed an improved tabu search algorithm, which is used to select valuable solutions from the initial solution set to construct a new solution set. Wilck and Cavalier [29] designed two hybrid algorithms to solve SDVRP based on the genetic algorithm, one is based on the shortest path genetic between unit customer demand and unit distance ratio. Berbotto *et al.* [30] first used granularity computing technology to solve SDVRP, and designed a random granular tabu search (RGTS) algorithm based on random granularity computing. They define the threshold of granularity calculation as the current remaining loading capacity of each vehicle and provide a variety of neighborhood operations. The important idea of the algorithm is that the current solution is based on the hierarchical probability of random selection of neighborhood operation. At the same time, RGTS allows neighborhood search to accept infeasible solutions that do not meet the vehicle loading capacity constraints. Silva *et al.* [31] adopted a new Perturbator SDVRP, which greatly improved the optimal solution. Yan [32] used an iterative local search algorithm and a three-stage tabu algorithm to solve SDVRP.

In the theoretical research and practical application of vehicle routing problems, scholars will consider some problem characteristics and conditional constraints, such as customers needing to obtain services within a specific period, or have pickup and delivery requirements at the same time. According to different problem characteristics and condition constraints, the SDVRP can be derived from a variety of types, including the SDVRP With Time Window (SDVRPTW), Heterogeneous Fleet Vehicle Routing Problem with Split Delivery (HFVRPSD).

A summary of the algorithm for solving SDVRP and its derivative problems is shown in Table 3. However, the actual geographic paths between request points are not the straight-line paths between the two points, and how to use the actual geographic paths buffers have not been considered.

III. PROBLEM DEFINITION AND MATHEMATICAL MODEL

A. PROBLEM DEFINITION

Let the graph $G = (V, E)$ serve as a model of a fully connected road network, where V is the finite set of nodes, modeling intersections, and $E \subseteq V \times V$ is the set of directed arcs modeling one-way roads between intersections. That is, $(n, n') \subseteq E$ if and only if there is a road that permits traffic to flow from intersection n to intersection n'. Pickups can be made at the origin nodes, $O \subseteq V$, and deliveries can be made at the destination nodes, $D \subseteq V$. There is no depot in the network, the vehicle leaves empty at the beginning of a route, and need not to return to the start location. The starting positions of vehicles of the third-party logistics enterprise is A, $V = A \cup O \cup D$. The distance d_{ij} between any two logistics nodes i and j is known. At present, there are K free transport vehicles and M orders to be transported in the transport network, where the starting position of each vehicle is known as $A_k \in A$, the rated load is known as Q_k, and the detention time is known as $Det^k_{transport}$. Each order includes a pickup node $P_m \in O$ and a delivery node $D_m \in D$ the quantity $q_m \in N^+$ of passenger finished vehicles to be transported and the order price W_m of the order m are known, and each order has a order generation time $T^m_{generate}$, a retention time $T^m_{retention}$ and the latest delivery time $T^m_{delivery}$ which generally is far greater than $T^m_{generate}$ in actual passenger finished vehicle logistics. It is necessary to formulate a reasonable transportation task and path planning to maximize the total profit of this scheduling.

B. MATHEMATICAL MODEL

1) NOTATIONS

a: COEFFICIENTS

- f^k_m: the unit loading cost of transport vehicle k transport order m, unit: ￥/ *vehicle* · *km*;
- e^k: no-load cost of transport vehicle k, unit: ￥/ *km*;
- F_k: fixed cost of transport vehicle k, unit: ￥/ *time*;
- d^k_m: total distance traveled by vehicle k after finishing the transportation of order m, unit: *km*;
- v_k: average driving speed of transport vehicle k, unit: *km /h*
- ST: scheduling period, unit: *day*
- Q_k: capacity of the transport vehicle k
- q_i: the demand of the node i, is q_i for the pick-up node, and -q_i for the delivery node.
- $Det^k_{transport}$: detention time of transport vehicle k
- $T^m_{generate}$: generation time of order m
- $T^m_{delivery}$: latest delivery time of order m

b: DECISION VARIABLES

$$x^k_{ij} = \begin{cases} 1, & \textit{transport vehicle k from node i to j} \\ 0, & \textit{otherwise} \end{cases}$$
$$\forall k \in A \quad \forall i \in O\ \forall j \in D$$

$$y^k_m = \begin{cases} 1, & \textit{order m translated by transport vehicle k} \\ 0, & \textit{otherwise} \end{cases}$$
$$\forall k \in A \quad \forall m \in O$$

$$q^m_{ij} = \begin{cases} 1, & \textit{order m is transported and travels from} \\ & \textit{node i to j} \\ 0, & \textit{otherwise} \end{cases}$$
$$\forall m = 1, 2, \ldots, M$$

- Q^k_i: load of transport vehicle k when leaving node i
- q^k_m: numbers of passenger vehicle of order m transported by transport vehicle k

2) MODEL OHFPDPTWSL

the following intermediate variables are defined to facilitate the description of the model:

Travel time of transport vehicle k from i to j:

$$t^k_{ij} = \frac{d_{ij}}{v_k} \quad \forall k \in A\ \forall i \in O\ \forall j \in D \tag{1}$$

Variable cost of transport vehicle k from node i to j:

$$c^k_{ij} = x^k_{ij}(e^k + f^k_i q^k_i)d_{ij} \quad \forall k \in A\ \forall i \in O\ \forall j \in D \tag{2}$$

Whether transport vehicle k participates in transportation:

$$y_k = \sum_{k=1}^{K}\sum_{m=1}^{M} y^k_m \tag{3}$$

Average transport vehicle detention time:

$$T_{avgDet} = \frac{\sum\limits_{k=1}^{K} Det^k_{transport}}{K} \tag{4}$$

TABLE 3. Solving algorithm summary of SDVRP and its derivative problem.

Model	Algorithm class	Author (year)	Algorithm
SDVRP	Exact algorithm	Dror *et al.* (1994)	Branch and bound
		Jin *et al.* (2007)	Cluster-path (two-stage method)
		Moreno *et al.* (2010)	Column generation and sectioning
		Archetti *et al.* (2014)	Branch and cut method (two types)
		Gschwind et al. (2019)	Stabilized branch-price-and-cut
	General heuristic	Belenguer *et al.* (2000)	Tangent plane method
		Campos *et al.* (2008)	Scanning algorithm
		Yan *et al.* (2015)	Two-stage method
	Hybrid heuristic	Chen *et al.* (2007)	C-W saving method, memory updating method
		Archetti *et al.* (2008)	Tabu search algorithm
		Jin *et al.* (2008)	Column generation method, delimited search algorithm
		Khmelev *et al.* (2015)	Variable neighborhood descent (VND), random tabu search, fast heuristic algorithm
		Haddad et al. (2018)	Large neighborhood search and branch-and-price
		Bortfeldt and Yi (2020)	local search algorithm,genetic algorithm and several construction heuristics
	Meta-heuristic	Archetti *et al.* (2006)	Tabu search algorithm
		Derigs *et al.* (2010)	Meta-heuristic algorithm based on neighborhood search
		Aleman *et al.* (2010)	Adaptive memory algorithm
		Aleman *et al.* (2010)	Tabu search algorithm
		Wilck *et al.* (2012)	Genetic algorithms (two types)
		Berbotto *et al.* (2014)	Granularity calculation
		Silva *et al.* (2015)	Iterative local search algorithm
		Xiong-hao *et al.* (2015)	Three-stage taboo algorithm
		Shi et al. (2018)	Particle swarm optimization
		Gu et al. (2019)	Adaptive large neighborhood search
SDVRPTW	Exact algorithm	Ceselli *et al.* (2009)	Column generation
		Archeui *et al.* (2010)	Improved branch-price-cut algorithm
		Salani *et al.* (2011)	Branch-pricing algorithm
		Luo *et al.* (2017)	Branch-price-cut
	Heuristic	Mcnabb (2015)	Improved heuristic algorithm
		Min et al. (2019)	Maximum-minimum distance clustering
HFVRPSD	Heuristic	Hemiig *et al.* (2012)	Heuristic algorithm
		Hertz *et al.* (2012)	Cluster-path heuristic
		Belfiore *et al.* (2013)	Scan search algorithm
		Kergosien *et al.* (2013)	Genetic Algorithm and Tabu Search Algorithm
		Chen *et al.* (2014)	Variable neighborhood search heuristic
		Wang *et al.* (2014)	Two-stage heuristic
		Wang *et al.* (2015)	Competition decision algorithm
		Salazar *et al.* (2015)	Branch-cut method
		Lee *et al.* (2015)	Large-scale neighborhood search
		Xia and Fu (2018)	Adaptive Tabu Search Algorithm

TABLE 4. Comparison between path buffer clustering and K-means clustering (the path buffer clustering outperforms 66% (37 out of 56) of the data).

Instance	NCC	Clustering Result ACA-OPB SC	K-means SC	Instance	NCC	Clustering Result ACA-OPB SC	K-means SC
LC101	10	**0.7351**	0.1058	LC201	3	0.0061	**0.2949**
LC102	10	**0.7276**	0.0685	LC202	11	0.0426	**0.0474**
LC103	10	**0.7225**	0.0775	LC203	13	0.0185	**0.0679**
LC104	11	**0.6457**	0.0618	LC204	11	**0.0472**	0.0414
LC105	10	**0.7282**	0.0788	LC205	14	**0.0417**	0.0036
LC106	10	**0.7296**	0.0692	LC206	7	**-0.0021**	-0.0171
LC107	10	**0.7251**	0.0377	LC207	14	**0.0457**	0.0262
LC108	10	**0.7327**	0.0477	LC208	4	0.0458	**0.1287**
LC109	10	**0.7376**	0.0609	LR201	8	0.0035	**0.0898**
LR101	14	**0.0561**	0.0003	LR202	16	**0.0366**	0.0235
LR102	20	**0.1115**	0.0154	LR203	6	0.0129	**0.0768**
LR103	20	**0.0988**	0.0323	LR204	16	**0.2295**	0.0420
LR104	9	**0.1721**	0.0754	LR205	16	**0.0531**	0.0516
LR105	9	**0.2178**	0.0094	LR206	16	**0.2394**	0.0485
LR106	15	**0.0798**	0.0528	LR207	15	**0.1872**	0.0603
LR107	11	**0.1619**	0.0860	LR208	17	**0.1442**	0.0765
LR108	11	**0.1619**	0.0775	LR209	18	**0.1312**	0.0267
LR109	6	0.0462	**0.1462**	LR210	15	**0.3040**	0.0655
LR110	6	0.0085	**0.1251**	LR211	14	**0.2867**	0.0464
LR111	6	-0.0292	**0.1040**	LRC201	29	**0.0272**	0.0159
LR112	2	0.0285	**0.3627**	LRC202	29	**0.0236**	0.0093
LRC101	2	0.0985	**0.3110**	LRC203	18	0.0161	**0.0270**
LRC102	3	-0.0860	**0.3448**	LRC204	24	**0.0683**	-0.0018
LRC103	3	0.1927	**0.3497**	LRC205	28	**0.0920**	-0.0016
LRC104	4	0.1233	**0.1703**	LRC206	24	**0.0799**	0.0091
LRC105	3	-0.0780	**0.1017**	LRC207	31	**0.0555**	0.0076
LRC106	5	0.0803	**0.1527**	LRC208	25	**0.0478**	0.0084
LRC107	5	-0.0645	**0.0388**				
LRC108	2	0.0904	**0.3481**				

Total no-load rate of transport vehicles:

$$\eta = 1 - \frac{\sum_{k=1}^{K}\sum_{i}^{i\in O}\sum_{j}^{j\in D}(d_{ij}x_{ij}^{k}(\sum_{m=1}^{M} q_{m}^{k}q_{ij}^{m}))}{\sum_{k=1}^{K}\sum_{i}^{i\in O}\sum_{j}^{j\in D} d_{ij}x_{ij}^{k}Q_{k}} \tag{5}$$

The model takes the maximum total profit as the objective as follow:

Total profit calculated by total amount subtract fixed cost, and then subtract variable cost:

$$W_1 = \sum_{m=1}^{M} W_m \sum_{k=1}^{K} y_m^k - \sum_{k}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M} c_{ij}^k - \sum_{k=1}^{K} F_k \sum_{m=1}^{M} y_m^k \tag{6}$$

Order timeout penalty cost:

$$W_2 = \sum_{k=1}^{K}\sum_{m=1}^{M} y_m^k P(T_{retention}^m + \frac{d_m^k}{v_k} - T_{delivery}^m) \tag{7}$$

Detention penalty cost of transport vehicle:

$$W_3 = \sum_{k=1}^{K} R(\frac{Det_{transport}^k + ST(1-y_k)}{T_{avgDet}}) \tag{8}$$

$$\text{Objective: } \max\ W_1 - W_2 - W_3 \tag{9}$$

$$\text{Subject to: } Q_j^k \geq Q_i^k + q_j - (Q_k + q_j)(1 - x_{ij}^k)$$
$$\forall i, j \in V,\ \forall k \in A \tag{10}$$

$$\sum_{k=1}^{K} q_m^k \leq q_m \quad m = 1, 2, \ldots, M \tag{11}$$

$$\sum_{k=1}^{K} x_{ij}^k = 0 \quad \forall i, j \in A \tag{12}$$

$$\sum_{j\in V, i\neq j} x_{ji}^k = \sum_{j\in V, i\neq j} x_{ij}^k \quad \forall i \in O \cup D\ \forall k \in A \tag{13}$$

$$\sum_{k=1}^{K} y_i^k \geq 1 \quad i = 1, 2, \ldots, M \tag{14}$$

$$\sum_{j\in V, j\neq i} x_{ij}^k = y_i^k \quad \forall k \in A\ i = 1, 2, \ldots, M \tag{15}$$

$$q_m^k \in N^+ \quad k = 1, 2, \ldots, K\ m = 1, 2, \ldots, M \tag{16}$$

Equation (9) is the objective function, which means the total profit subtract the penalty value of exceeding the time window, and then subtract the detention penalty value of the transport vehicle. Equation (10) is the capacity constraint which means the number of passenger vehicles transported

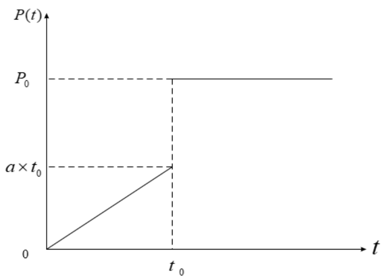

FIGURE 3. Order timeout penalty function.

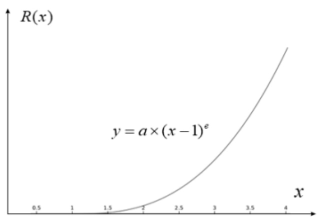

FIGURE 4. Vehicle detention penalty function.

by a transport vehicle must not exceed the maximum capacity limit of this transport vehicle. Equation (11) indicates that the sum of the passenger finished vehicles of an order transported by a transport vehicle must be less than or equal to the total quantity of this order. Equation (12) indicates that the transport vehicle must not be allowed to travel from the starting point to the starting point. Equation (13) means that the transport vehicle must leave after reaching the demand node (the pickup node or delivery node). Equation (14) means that each demand node is served at least once. Equation (15) means that the same transport vehicle must arrive at the delivery node after leaving the pickup node of the order. Equation (16) is a constraint of non-negative integers for decision variables which means numbers of passenger vehicles of order m transported by transport vehicle k.

This article uses the non-linear functions shown in Fig. 3 and Fig. 4 as the order overtime penalty function and the vehicle detention penalty function according to the actual application scenario of the enterprise. In Fig. 3, $[0, t_0]$ indicates that the order penalty value increases linearly with time, P_0 represents the maximum penalty value of the order. In this article, a takes 100, t_0 takes 30 days, and P_0 takes 10,000 ¥. The detention cost of the passenger finished vehicle in Fig. 4 increases with the increase of detention time which the purpose is to prevent the order from having not been served for a long time. In this article, a is taken as 100.

IV. SOLUTION APPROACH

Geographic Information System (GIS) is widely used in scientific investigation, resource management, property management, development planning, and other aspects. In recent years, some scholars have used GIS to solve some hot issues in logistics [53]. Buffer Area (BA) is a kind of influence scope or service scope of geospatial target, specifically refers to a certain width of multilateral automatically established around the point, line, and surface entity, which is mathematically expressed as $Bi = (x : d(xi, Oi) \leq R)$, specifically can be divided into point buffer area, line buffer area, and surface buffer area.

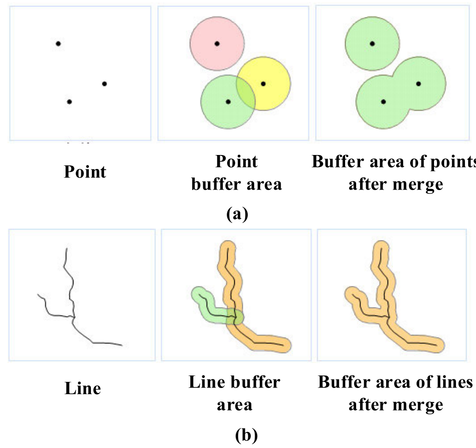

FIGURE 5. Concept of buffer area.

As shown in Fig. 5 (a), the point buffer area is a circular area generated by taking the point object as the center and the given buffer distance as the radius. As shown in Fig. 5 (b), the line buffer area is the normal direction of the object along the line, which is a closed area formed by translating two lines to two sides of the line object for a certain distance and joining the smooth curve formed at the end of the line.

This article proposes a new path buffer clustering and load split route planning hybrid parallel algorithm (PBCLSR-PHPA) to solve the problem. Firstly, the operator of the adaptive clustering algorithm based on the order path buffer (ACA-OPB) is used to cluster the orders, and then splits the requirements and planning the route for each cluster separately. Because each cluster is independent of each other, all steps of route and split are performed by different processors of multi-core CPU in parallel to speed up the whole process. The overall framework of the algorithm is shown in **Algorithm 1**.

A. PATH BUFFER CLUSTERING

The adaptive clustering algorithm based on the order of actual geographic path buffer is called path buffer clustering (PBC) and includes two steps: order grouping and adding vehicles to the group to form a cluster. Firstly, different buffer sizes are set according to the order clustering type, and orders with overlapping buffer areas are grouped into one group. Then the size of all the transport vehicle locations buffer δ (point buffer of the transport vehicle's start location) is set, and the transport vehicles with δ buffer overlapping with a point buffer of the pickup location in a group are added to this group to form a cluster. The schematic diagram of the clustering algorithm for transportation orders based on path buffer is shown in Fig. 6.

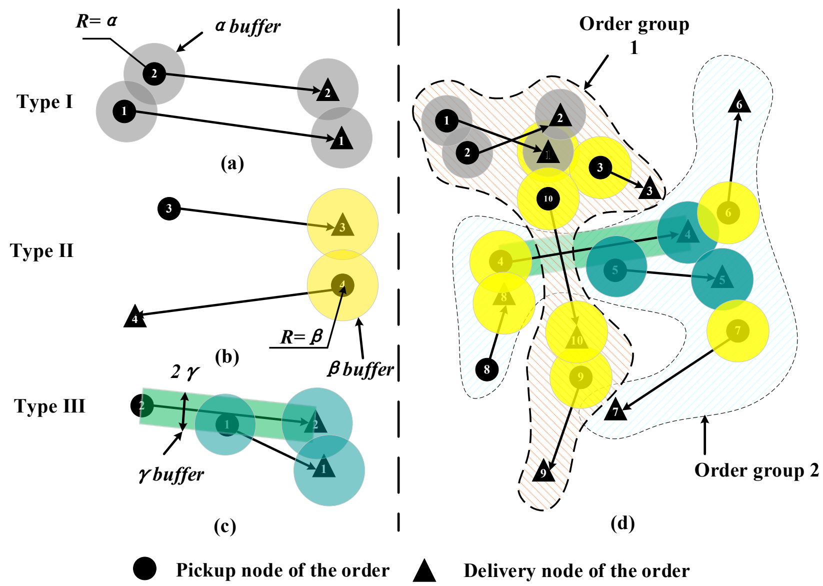

FIGURE 6. Schematic diagram of clustering algorithm for transportation orders based on path buffer.

Algorithm 1 Framework of the Algorithm

Input: the request list A,
the transport vehicle list C
demand split type st
Output: Optimized results RS
$results \leftarrow cluster(A, G)$ (**Algoritlmi2**)
for G *in results* **do**
// The following operations are performed in parallel
// The ForkJoin framwork in Java is used in this article
if $st ==$ *'multi-mark split'* **then**
$R \leftarrow route(G)$ (**Algorithm 4, 5**)
$S \leftarrow MMS(R)$ (**Algorithm 3**)
end
else
$S \leftarrow unitSplit(G)$ // (**section 4.2**)
$R \leftarrow route(S)$ (**Algorithm 4, 5**)
end
end

There are three different types of buffer clustering:

1). Type I clustering: as shown in Fig. 6 (a), the α buffer (the radius of buffer size is α) of the pickup node and the delivery node of two orders is overlapped respectively, the clustering buffer parameter is α.

2). Type II clustering: as shown in Fig. 6 (b), the β buffer (the radius of buffer size is β) of the delivery node of the first order and the β buffer of the pickup node of the second order is overlapped, the clustering parameter is β (generally $\beta > \alpha$).

3). Type III clustering: as shown in Fig. 6 (c), the α buffer of the pickup node of one order and the γ buffer (the width of the line buffer area of the transportation route is 2γ) of another order is overlapped, and at the same time, the β buffer of the delivery node of the two orders are overlapped.

According to the above steps, the order can be divided into several groups, a schematic diagram of order clustering results including two order groups is shown in Fig. 6 (d). The cluster generated according to the above method may have the situation of repeated usage of transport vehicles (that is the same transport vehicle may be added to several different cluster), and need remove duplicates according to the following principles:

1) First, the distance is preferred, that is, the transport vehicle with the shortest distance away from an order pickup point in a cluster is added to this cluster;

2) Second, the ratio of the number of vehicles in the cluster to the number of orders is considered, that is, the transport vehicle will be added to the cluster with a smaller ratio;

3) Finally, the transport vehicle will be added to a cluster randomly. The pseudo-code of the clustering algorithm is shown in **Algorithm 2**.

B. MULTI-MARK SPLIT

The existing request split methods mainly include the basic split method, greedy split method and proportional split method [52]. According to the characteristics of the pickup and delivery problem, this article adopts two kinds of split methods: unit split and multi-mark proportion split.

1) UNIT SPLIT

Assuming that the demand at customer point i is q, it can be split into q customers with a demand of 1. These customers have a pickup node and a delivery node, and the distance between the pickup (delivery) node and the pickup(delivery) node is 0. In this way, we can remove the constraint that

---

Algorithm 2 Path Buffer Clustering (PBC)

Input: the request list A, α, β, γ
Output: clustering results CR
$n \leftarrow$ *size of* A
Adjacency $\leftarrow$ *empty list with size* $n * n$ *and de faut value* 0
for *i (0, n)* **do**
  for *j in (0, n)* **do**
    if $i\ ! = j$ **then**
      $r_i \leftarrow requestes[i]$
      $r_j \leftarrow requestes[j]$
      if r_i *and* r_j *satisfy the clustering conditions* **then**
        $Adjacency[i][j] \leftarrow 1$
      end
    end
  end
end
The number of connected graphs in Adjacency is the number of clusters

---

the requirements can be split and convert it to a heterogeneous fleet pickup and delivery problem with time windows (HFPDPTW) for solving.

2) MULTI-MARK SPLIT

Multi-mark split algorithm is a kind of split algorithm that is similar to the traversal algorithm used to split the optimized route, but compared with traversing the whole route, the complexity of the multi-mark algorithm is much lower because it visits fewer edges in the network. The multi-mark algorithm scans every node of the pre-optimized route in turn, and inserts marks for each node on the route based on the optimal marks of the previous node. A mark of a node describes the status of a transport vehicle that will travel through this node. And as shown in Fig. 7 (a), a mark is also an entity that has many attributes such as mark no. (MN), nearest transport vehicle no. to this node (TN), distance from the node to this transport vehicle (ND), rated capacity of this transport vehicle (RC), the total quantity of passenger finished vehicle will be transported by this transport vehicle to this node (TQ), a total distance of that this transport vehicle will travel to this node (TD), nodes of this transport vehicle has traveled through in sequence (NS). New marks are only generated on pickup nodes, and marks on delivery nodes are inherited from the previous node of which the value of *TQ*, *TD*, and *NS* will be changed. The demand of pickup nodes that exceeds the capacity constraint of transport vehicles will be split and transported by a new transport vehicle. Because there may be many marks on the same task node, this split algorithm is called a multi-mark algorithm.

The steps of the multi-mark proportion splitting algorithm as follows:

i). Add new marks

Traverse each pickup node N_i in the network. If there is an empty transport vehicle near the task node and the demand of this node is not greater than the rated capacity of the nearest transport vehicle, add a new mark M_j, and the values of the attributes of this mark can be set as below:

$\mathrm{MN} = j \quad j = 1, 2, 3 \ldots$

TN is the transport vehicle no.

ND is the distance from this transport vehicle to the current task node.

RC is the rated capacity of this transport vehicle.

$TQ_j = min\,(C_i, RC)$ where C_i is the demand of the current task node.

$TD_j = \mathrm{ND}$

$\mathrm{NS} = i$

If the demand of this node C_i is larger than the rated capacity of the nearest transport vehicle RC, the demand of this node will be split as $RC \times R$ and $C_i - RC \times R$, and repeat the above steps.

As the mark no.1 on the first node 3(+4), the mark no.2 on the second node 2(+6), and the no.3 mark on the fourth node 1(+10) shown in Fig. 7 (b), they are all marks on the pickup node, and there is only one transport vehicle near each node no.1, no.2, and no.3.

ii). Inherit marks

Traverse each node N_k in the network, and do as follows:

If the node is a delivery node, inherit all the marks of the previous node, and the attributes of the mark can be changed as below:

MN, *TN*, *ND*, and RC keep unchanged.

NS, TQ, and *TD* can be set as below:

$$NS_j = NS_{j-1} \cup k \quad k = 1, 2, 3 \ldots$$
$$TQ_j = \begin{cases} TQ_{j-1} & \textit{If } NS_j \textit{ contains the pickup node of } node_k \\ TQ_{j-1} - C_i & \textit{Otherwise} \end{cases}$$
$$TD_j = TD_{j-1} + L\,(k-1, k)$$

where C_i is the demand of the current task node, $L(k-1, k)$ is the distance between task node $k - 1$ and k.

As marks on the third node 2(−6), the marks on fifth node 3(−4), and the marks on the sixth node 1(−10) shown in Fig. 7 (b), they are all marks on the delivery node.

If the current node is a pickup node, inherit all the marks of the last node ahead of this node and change the value of *TQ*, *TD,* and *NS*, keep other attributes unchanged.

TQ_j can be set as $TQ_j = min(TQ_{j-1} + C_i, C_r)$, where C_i is the demand of the current task node, $C_r = RC \times R$, $R \in (0, 1]$, and R is the proportion parameter, RC is the rated capacity of the transport vehicle of mark j.

TD_j can be set as: $TD_j = TD_{j-1} + L\,(i-1, i)$ where $L(i-1, i)$ is the distance between task node $i - 1$ and i.

NS_j can be set as: $NS_{j-1} \cup k$.

If the demands of this node are larger than the remaining capacity of the inherited transport vehicle, the exceeded demands will be transported by a new transport vehicle near

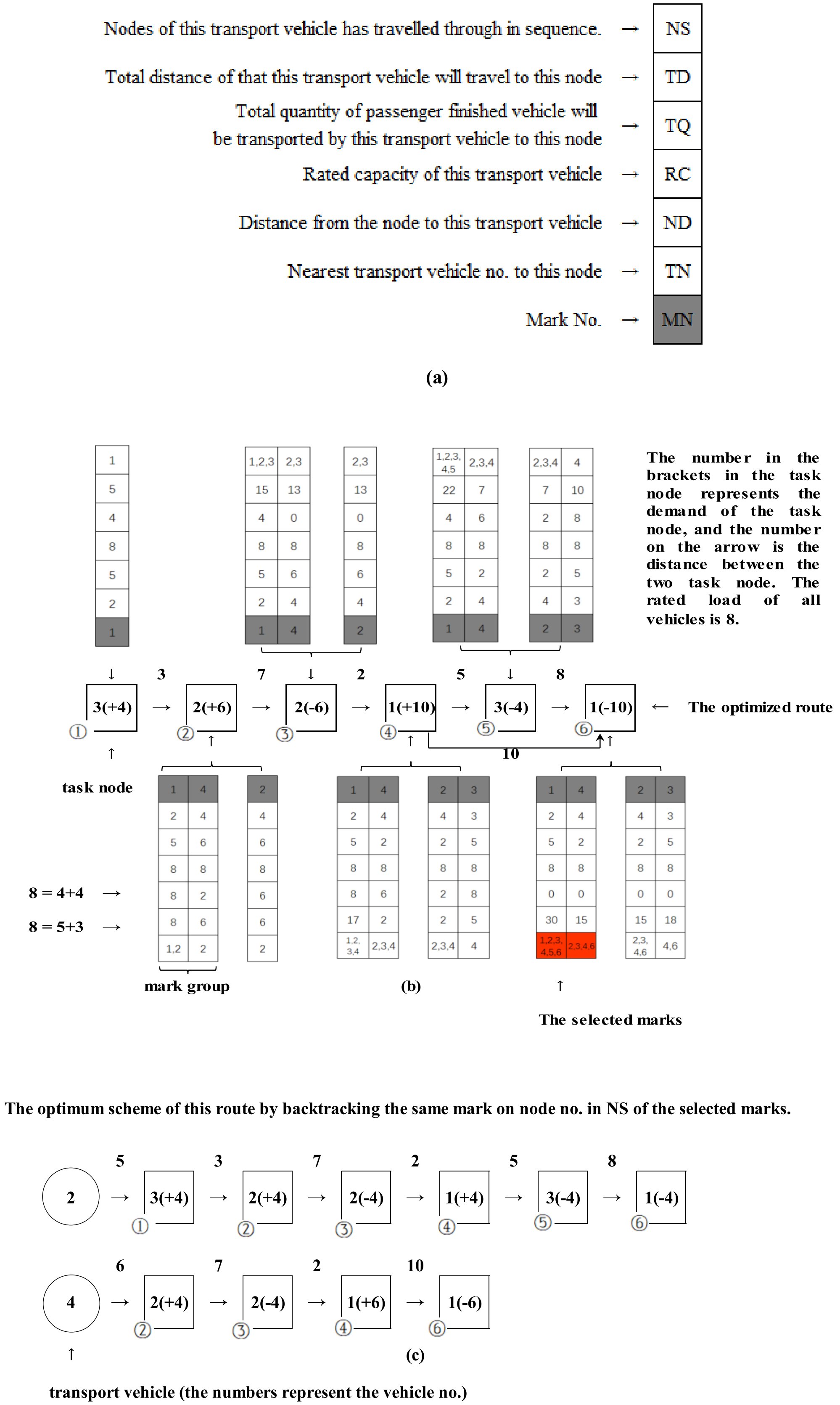

FIGURE 7. Schematic diagram of the multi-mark proportional split algorithm.

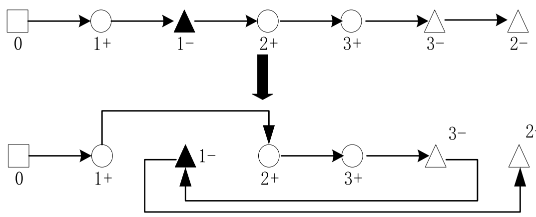

FIGURE 8. Relocate node in one route: one pickup node or a delivery node is removed to be reinserted in the best position of the same route.

this node, that is to say, a new mark will be created. And the method to create the new mark as step 1.

As the mark no.1 and no.4 on the second node 2(+6), the mark no.2 on the fourth node 1(+10) shown in Fig. 7 (b) are marks on delivery nodes inherit from the last node ahead of them.

Thus, the marks on the last node of the pre-optimized route are all marks of this route.

3) SELECT AND BACKTRACK MARKS

Select marks on the last node that their *NS* can include all nodes in the pre-optimized route and the summation of their *TD* is the shortest as the final marks. We can get the optimum scheme of this route by backtracking the same mark on node no. in *NS* of these marks. The detailed implementation process of the multi-mark split algorithm is shown in **Algorithm 3**.

As shown in Fig. 7 (b), the mark no.1 and no.2 are the selected marks because their *NS* have included all the nodes from no.1 to no.6 and the summation of distance traveled by all transport vehicles is the shortest 45. By backtracking the same mark on a node no. in *NS* of mark no.1 and no.4, the last optimum scheme of this route is shown in Fig. 7 (c).

C. ROUTE PLANNING

Since the number of orders in each cluster is small after clustering, the improved tabu search algorithm is used to plan the route. This article implements four neighborhood search operators to generate a new solution. These moves are illustrated in Fig.8-11. As shown in the figures, squares represent transport vehicles while circles represent pickup nodes and triangles denote delivery nodes.

The first two operators involve the transformation of nodes, and the next two involve the transformation of edges. All these four operators are to generate better new solutions for path optimization. During each iteration, randomly select one of the four neighborhood search operators to generate one/two neighborhoods. If the neighborhood is an infeasible solution, skip it. The pseudo-code for route planning is shown in **algorithm 4** and the pseudo-code of neighborhood generation is shown in **algorithm 5**.

D. TERMINATION CONDITION

Using the convergence termination method as the termination condition of route planning, detect the change of the objective function value with iteration. If the change of the objective function value satisfies the convergence condition,

Algorithm 3 Multi-Mark Split(MMS)

```
Input: the pre-optimized route por, the transport vehicle lists C
Output: split results SR
nodes ← tasknodes of por
// Create Marks
marks ← {}
for node in nodes do
    mn ← 0 // Quantity of mark or mark Group of the node
    if node is a pickup node then
        nv ← find nearest transport vehicle to node from C
        if nv! = null then
            capacity ← node.capacity
            // Add New Marks
            if capacity ≤ nv.capacity then
                mark ← AddNewMarks(marks, nv, node)
                marks[node][mn++] ← mark
            end
            else
                markGroup ← AddNewMarks(marks, nv, node)
                marks[node][mn++] ← markGroup
            end
        end
        if node is not the first node then
            // Inherit Marks
            mark[node][mn++] ← InhcritMarks(marks, node)
        end
    end
        // Inherit Marks
        marks[node][mn++] ← InheritMarks(marks, nodc)
    end
end
FM ← marks of the last node
SR ← find the marks that can travel all of the nodes of por and with
the shortest summation of travel distance from FM
```

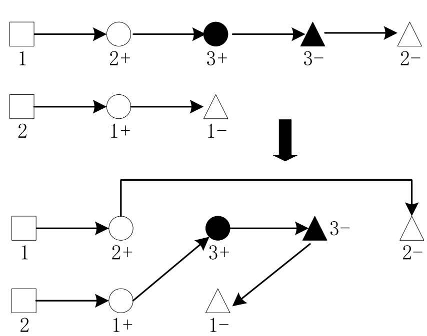

FIGURE 9. Relocate node between routes: one request is removed from one route and reinserted in the best position of another route.

Algorithm 4 Find Shortest Route (FSR)

Input: the request list A, the transport vehicle list C
Output: Solution s
Randomly generate a solution i, and evaluate it
$f(i)s \leftarrow i, k \leftarrow 0, H \leftarrow \{\}$
while *not stop* **do**
  // generate neighbors by the four operators
  $E \leftarrow GNS(i, H)$
  $i \leftarrow SelectBestSolution(E)$
  Update the tabu list H
  if $f(i)$ *better than* $f(s)$ **then**
    $s \leftarrow i$
  end
  $k \leftarrow k + 1$
end

Algorithm 5 Generate Neighborhood Solution (GNS)

Input: the solution i, the tabu list H
Output: Solution list E
$itr \leftarrow 0$, $R \leftarrow$ Initializing an array with size N
N is the number of new solutions
while $itr \leq N$ **do**
  $t \leftarrow rand(0, 4)$
  $s \leftarrow$ *Select a solution from H randomly*
  if $t = 0, 2$ **then**
    $R[itr] \leftarrow$ *Use i to generate a neighbor by the first and third VNS operator illustrated in Fig. 8 and Fig. 10*
    $itr \leftarrow itr + 1$
  end
  if $t = 1, 3$ **then**
    $R[itr], R[itr + 1] \leftarrow$ *Use i and s to generate two neighbors by th second and fourth VNS operator illustrated in Fig.9 and Fig.11*
    $itr \leftarrow itr + 2$
  end
end

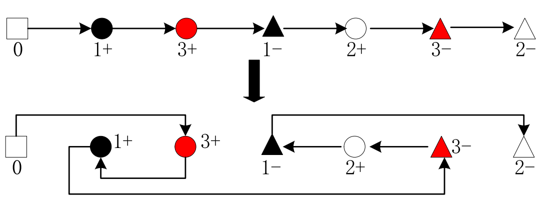

FIGURE 10. Relocate edge in one route: two requests are exchanged in the same route.

the optimization ends. The formula of convergence termination method is as follows:

$$|\frac{f_{k+t} - f_k}{f_k}| \leq \varepsilon \tag{17}$$

In the above formula, f_k represents the optimal objective function value at the kth iteration, f_{k+t} represents the optimal objective function value at the number $k + t$ iteration and ε is an arbitrarily small positive number (this article takes 0.001).

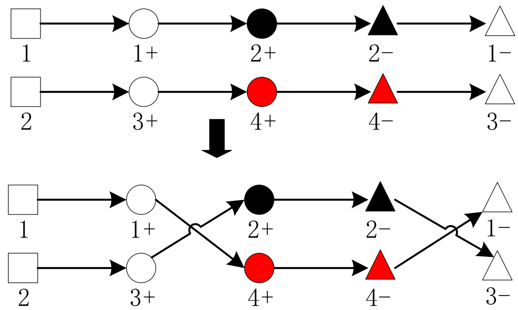

FIGURE 11. Relocate edge between routes: two requests are exchanged between two routes.

V. NUMERICAL EXPERIMENTS

We use java to code programs and run them in the win10 operating system. And the system configuration is AMD 3600/4.2Ghz.

The algorithm java program source code of this article can be obtained freely from https://gitee.com/bupt_htl/pdptw.

A. VALIDATION ON THE REVISED INSTANCES

1) INSTANCE INTRODUCTION AND PARAMETER SETTING

The main aim of this article is to propose new algorithms for real-world OHFPDPTWSL instances. Because there are no benchmark instances that can be used directly for the OHFPDPTWSL problem. Li and Lim's benchmark revised from Solomon's benchmark instance for pickup and delivery problem with time windows which are widely used [19], [54], which can be used to verify the model and algorithm of this article after a simple revision. These data are divided into three categories: clustering data (LC), semi-clustering data (LRC), and discrete data (LR), and each category of data is divided into two groups. The instance requires all transport vehicles to start from a fixed distribution center with the same capacity. Therefore, the instances are revised as follows:

1) Remove the distribution center constraint in the original instance;

2) Calculate the upper and lower boundaries of X and Y axes of each group of data respectively: X_m, X_M, Y_m and Y_M;

3) Set the parameter λ to represent the proportion between the number of transport vehicles and the number of orders, then the number of transport vehicles added to each group of data is $\lambda \times n_o$,where n_o is the number of orders

4) The coordinates of an added vehicle are:

$$X = rand\,(0, 1) \times (|X_M - X_m|)$$
$$Y = rand(0, 1) \times (|Y_M - Y_m|);$$

5) The capacity of the transport vehicle is randomly selected as 5, 10, or 15.

The schematic diagram of the method to improve the instance is shown in Fig. 12. The red triangle represents the pickup point of the original order, the black triangle

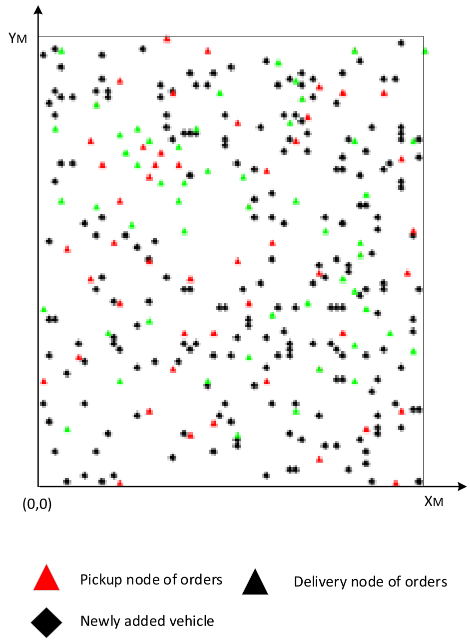

FIGURE 12. **Schematic diagram of instance improvement.**

represents the delivery point of the original order, and the black diamond represents the newly added vehicle. For the third-party logistics platform, the quantity of transport vehicles is generally large enough, so the parameter λ in this article is set to 2 which can ensure that at least one transport vehicle can be found near each pick-up node (VSC). So in each group of data in the revised benchmark instances, there are many starting point data of transport vehicles, and these revised data set can be obtained from http://www.301lib.com/pdpsl.

The four parameters of the algorithm (α, β, γ, and δ) in this article are very important to the experimental results, which affect the value of the objective function. Some experiments by taking different values of these parameters show that these parameters are closely related to the geographical distribution of orders and transport vehicles, but not to the capacity of transport vehicles and the requests of customers. According to the experiment, the following empirical formulas can be summarized:

$$\alpha = \frac{2\sum_{i=0}^{n}\sum_{j=1}^{n}\alpha_{ij}}{kn(n-1)(X_M - X_m + Y_M - Y_m)} \tag{18}$$

$$\beta = \frac{2\sum_{i=0}^{n}\sum_{j=1}^{n}\beta_{ij}}{kn(n-1)(X_M - X_m + Y_M - Y_m)} \tag{19}$$

$$\gamma = \frac{2\sum_{i=0}^{n}\sum_{j=1}^{n}\gamma_{ij}}{kn(n-1)(X_M - X_m + Y_M - Y_m)} \tag{20}$$

$$\delta = \frac{2\sum_{i=0}^{m}\sum_{j=1}^{n}\delta_{ij}}{km(n-1)(X_M - X_m + Y_M - Y_m)} \tag{21}$$

where m represents the number of vehicles, n represents the number of orders, k is a constant (10 is taken in this article), α_{ij} represents the distance between the pickup node of orders i and j, β_{ij} represents the distance between the pickup node of orders i and the delivery node of orders j, γ_{ij} represents the vertical distance from the pickup node of order i to the line through pickup node and delivery node of order j, and δ_{ij} represents the distance between transport vehicle i and the pickup node of order j.

2) PERFORMANCE METRICS OF CLUSTERING EFFECT

The path buffer clustering method in this article is a kind of unlabeled clustering method, it needs to use the indicators such as compactness and separation to evaluate the clustering effect as discussed in [55]. In this article, the Silhouette Coefficient (SC) is used to evaluate the clustering effect. The mathematical expression is as follows:

$$sc = \frac{b-a}{\max(a,b)} \tag{22}$$

$$a = \frac{\sum_{i\in S}\sum_{j\in S, i!=j} d(i,j)}{size(S)-1} \tag{23}$$

$$b = \frac{\sum_{i\in S}\sum_{j\in T} d(i,j)}{size(T)} \tag{24}$$

where a is the average distance between the sample and other points in the same cluster, and b is the average distance between the sample and other points in the next closest cluster. S and T are sample sets of the current cluster and nearest cluster to the current cluster respectively. The value of the silhouette coefficient is between $[-1, 1]$. The closer the SC to 1, the higher the internal compactness among clusters, the better the clustering effect. The average of the silhouette coefficients of all points is the total silhouette coefficients of the clustering results.

The distance between sample i and j is calculated as follows:

$$d(i,j) = \sqrt{((s_{ij} + e_{ij})/2)^2 + (\min(se_{ij}, se_{ji}))^2} \tag{25}$$

where s_{ij} represents the distance between the pickup node order i and j, e_{ij} represents the distance between the delivery node of order i and j, se_{ij} represents the distance between the pickup node of order i and the delivery node of order j, and se_{ji} represents the distance between the pickup node of order j and the delivery node of order i.

To show the result of K-*means* clustering compared to the path buffer clustering, its parameters are set as follows:

1) The clustering distance is calculated according to formula (19)
2) the number of clustering centers is set to be the same as the path buffer clustering algorithm.

The clustering effect was verified by these revised instances with 50 requests. The results are shown in Table 4, where NCC represents the number of clustering centers. The experimental results show that the path buffer clustering algorithm outperforms 66% (37 out of 56) of the data.

3) EFFECT OF SPLIT PROPORTION

The key parameters of the algorithm are of great significance to its practical operations [56], [57]. The Split Proportion (SP)

TABLE 5. Comparison of transport vehicle usage, total distance (*TD*) and percentage of loading rate under different split proportions (with the increase of split proportion, the loading rate of vehicles (*LR*) tends to increase and the quantity of transport vehicles used tends to decrease).

R	LC101			LR101			LRC101		
	VN	TD	LR	VN	TD	LR	VN	TD	LR
0.50	181	16093.12	78.56%	236	28460.10	84.66%	237	37056.98	84.00%
0.55	180	16031.36	81.73%	244	29883.80	91.22%	213	33822.38	85.97%
0.60	169	15080.32	80.38%	222	27666.54	86.68%	207	33465.08	87.30%
0.65	174	15507.18	86.52%	199	24598.80	81.42%	192	30780.42	84.84%
0.70	157	13971.38	81.70%	212	25784.89	90.93%	191	30109.93	88.41%
0.75	148	13210.43	81.08%	203	23891.35	91.22%	190	29015.17	92.24%
0.80	157	13939.43	89.92%	196	19319.98	92.89%	171	21845.56	87.46%
0.85	140	12453.78	84.59%	174	17607.19	86.85%	170	22273.80	91.49%
0.90	141	12523.75	89.72%	180	20473.37	94.73%	161	23763.47	91.55%
0.95	131	11672.43	88.38%	164	20911.87	91.20%	165	27299.40	94.00%
1.00	133	11851.11	95.05%	160	21912.53	94.34%	157	27746.67	96.00%

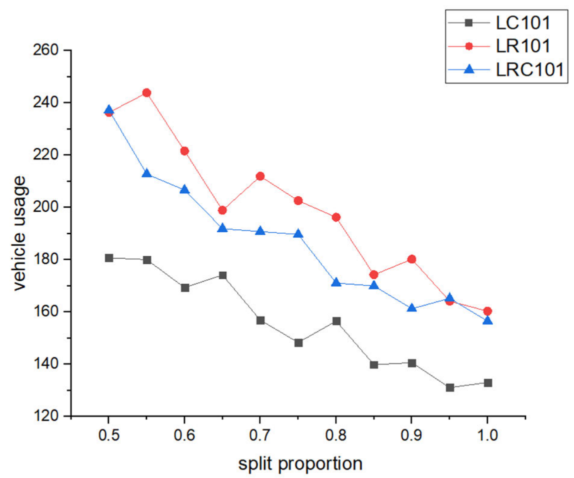

FIGURE 13. Comparison of the transport vehicle usage under different split proportions.

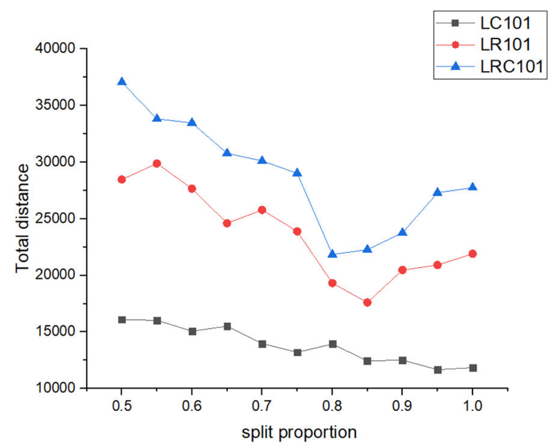

FIGURE 14. Comparison of time distance under different split proportions.

is an important parameter in this article. The clustering data (LC101), semi-clustering data (LRC101), and discrete data (LR101) is used to explore the optimal split proportion. The initial solution is constructed using Solomon's insertion algorithm, and the length of the tabu list is set to 10. The split proportion is set to 0.5 - 1.0 with the step of 0.05.

As shown in Table 5 and Fig. 13-15, where *R* represents split proportion, *VN* represents the number of transport vehicles used, and *LR* represents the percentage of loading rate. The experimental results show that with the increase of split proportion, the loading rate of transport vehicles tends to increase and the number of transport vehicles used tends to decrease, while the total distance tends to decrease firstly and then to increase quickly which means the objective value increase firstly and then decrease quickly, and the total distance is the smallest when the split proportion is about 85%, so we set the split proportion to 85% in following experiments.

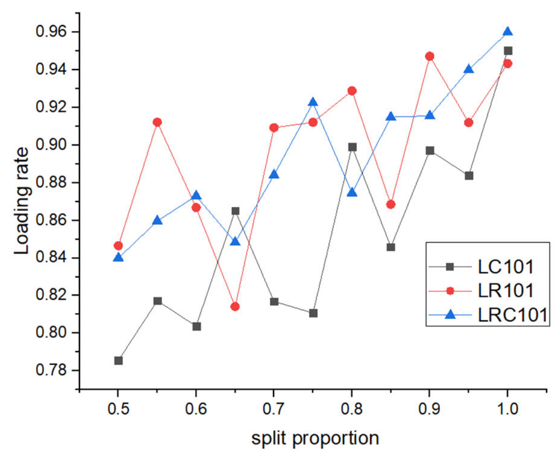

FIGURE 15. Comparison of loading rate under different split proportions.

4) EXPERIMENTAL RESULTS ON REVISED INSTANCES

These revised instances were used to verify the algorithm in this article. The split proportion of the multi-mark split algorithm is set to 0.85, and the degree of concurrency is set to 3. To facilitate comparison with the results of the benchmark instances, we involve all data in optimization. In this article, we only list results with 50 and 300 requests, and more results are given in the attachment.

TABLE 6. Experimental results with 50 requests after 10 runs (Multi-mark split has advantage in running time, unit split has advantages in transport vehicle usage (44 out of 56), total distance (43 out of 56), and loading rate (34 out of 56)).

	Unit split												Muti-mark split								
Instance	VN			TD			LR			RT			VN			TD			LR		
	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*
LC101	155	161.21	7.24	**13801.08**	13888.45	375.20	**77.12%**	80.67%	0.03	6.80	6.96	0.06	**132**	135.98	5.12	14189.81	14164.82	126.03	66.69%	65.28%	0.02
LC102	**143**	149.13	7.15	11849.19	12426.73	355.18	**72.12%**	72.04%	0.02	6.64	6.83	0.37	157	162.36	4.73	**11312.58**	11482.14	194.70	66.74%	66.66%	0.02
LC103	**157**	160.63	7.34	**13953.22**	14053.01	658.73	**79.68%**	82.25%	0.01	6.49	6.65	0.24	165	164.53	3.10	14931.53	15686.01	310.31	67.73%	68.52%	0.03
LC104	**141**	149.87	3.09	12133.64	12349.84	643.52	**75.69%**	76.84%	0.03	28.14	28.93	1.56	156	162.89	4.15	**11900.27**	12046.63	479.48	72.35%	73.84%	0.02
LC105	**150**	150.75	8.29	**12947.89**	13374.01	412.03	**81.31%**	82.90%	0.01	3.66	3.73	0.15	159	162.98	4.59	14059.31	14171.87	498.95	75.83%	74.97%	0.03
LC106	**144**	141.15	7.72	**11528.85**	11526.71	409.87	81.30%	83.64%	0.03	2.68	2.71	0.08	152	159.70	2.87	12601.77	12554.29	500.26	**82.80%**	84.70%	0.04
LC107	**162**	166.16	5.70	**15421.47**	15787.40	475.36	**83.26%**	85.04%	0.03	8.50	8.66	0.16	168	170.79	6.41	18571.81	18655.59	755.55	77.69%	79.18%	0.02
LC108	158	160.03	5.02	**14912.18**	14945.51	698.09	**82.48%**	84.32%	0.02	8.59	8.77	0.35	**141**	146.28	6.01	15003.60	14832.74	501.37	68.67%	69.46%	0.02
LC109	**169**	174.45	4.89	**17061.02**	17462.29	583.13	**77.49%**	79.90%	0.03	13.72	13.86	0.60	190	192.86	3.63	18132.41	18774.86	590.27	76.03%	76.70%	0.02
LR101	**193**	195.70	2.47	**15073.42**	15259.81	653.86	**78.82%**	79.34%	0.04	4.44	4.64	0.16	202	208.07	5.67	16064.63	16350.27	309.11	71.61%	71.51%	0.02
LR102	189	190.59	5.85	14863.78	15235.53	312.81	**72.11%**	74.16%	0.03	7.28	7.50	0.30	**182**	186.36	4.52	**13929.55**	14401.56	381.63	66.35%	67.12%	0.01
LR103	**177**	190.24	3.75	**15238.33**	15655.79	339.48	80.65%	81.50%	0.03	14.62	14.86	0.60	193	197.74	4.69	16260.26	15927.74	494.64	**83.53%**	85.97%	0.03
LR104	**184**	191.16	4.14	**15394.59**	15723.35	853.67	**71.53%**	72.26%	0.05	28.76	29.04	1.23	191	197.67	5.22	16468.57	16745.76	347.53	68.78%	71.73%	0.01
LR105	**189**	194.65	6.37	**16637.72**	16653.14	649.23	72.75%	73.88%	0.04	11.09	11.27	0.62	202	207.15	3.41	18743.60	18908.66	897.47	**81.18%**	82.11%	0.02
LR106	174	178.75	3.62	**14301.58**	14406.05	340.80	79.37%	77.76%	0.01	11.12	11.51	0.76	**171**	172.85	3.98	15203.39	15736.17	299.61	**80.26%**	80.93%	0.02
LR107	**160**	160.04	4.28	**14914.45**	15389.01	939.75	**74.34%**	77.15%	0.02	24.17	24.69	0.97	167	172.45	5.37	16656.04	17233.94	339.31	65.70%	67.40%	0.02
LR108	**176**	180.38	6.25	**15760.03**	16407.15	437.56	**77.00%**	78.45%	0.04	19.91	20.31	1.00	201	208.04	4.20	15891.07	15842.32	572.51	65.21%	66.02%	0.03
LR109	**191**	193.58	6.53	**16605.60**	17543.82	606.01	**80.06%**	82.82%	0.04	15.65	15.58	0.45	194	197.73	7.13	20753.00	21163.01	875.76	76.92%	77.30%	0.03
LR110	**171**	174.19	9.55	**15022.01**	15600.70	399.05	81.91%	80.78%	0.02	15.83	16.15	0.54	186	192.72	4.77	15043.40	15283.58	524.66	**83.69%**	85.29%	0.02
LR111	**151**	158.59	4.87	13438.41	13494.72	631.47	**83.77%**	84.85%	0.03	16.01	16.69	0.87	167	173.35	8.36	**12062.45**	12350.43	544.67	79.92%	82.14%	0.02
LR112	**171**	168.87	5.29	**16343.02**	16140.04	454.98	83.82%	85.34%	0.05	12.99	13.35	0.39	191	197.91	4.59	17305.09	17658.86	524.86	**89.49%**	89.13%	0.05
LRC101	**190**	190.81	8.66	**19247.64**	19142.30	584.24	83.88%	87.34%	0.02	9.96	10.05	0.36	229	238.45	8.50	20218.89	20971.01	417.59	**88.44%**	86.96%	0.02
LRC102	192	200.21	8.39	**20147.11**	20138.54	511.01	78.49%	79.27%	0.03	10.71	11.03	0.34	**179**	185.80	4.55	20256.95	20704.96	522.59	**83.79%**	83.90%	0.02
LRC103	**180**	183.96	6.28	**18307.54**	18850.15	391.08	73.18%	75.60%	0.02	22.30	22.37	0.72	211	215.89	8.04	20866.96	21463.49	792.71	**78.54%**	80.84%	0.02
LRC104	**200**	202.50	10.01	19367.79	20587.11	805.69	77.73%	81.19%	0.03	17.17	17.52	0.41	225	237.18	8.88	**17999.88**	18515.46	664.45	**83.68%**	82.80%	0.04
LRC105	**192**	195.83	6.11	**20428.05**	20411.01	617.73	**82.29%**	85.36%	0.02	12.05	12.35	0.60	196	200.67	7.84	21262.70	21202.11	670.80	81.48%	83.07%	0.02
LRC106	**184**	186.91	8.07	**21590.39**	22194.82	698.70	**81.71%**	86.46%	0.02	23.83	24.91	0.66	210	208.67	3.65	23434.82	24550.85	563.01	81.31%	83.26%	0.02
LRC107	193	195.55	7.96	20257.52	21040.69	1071.94	70.29%	70.38%	0.02	28.31	27.96	1.48	**182**	185.96	5.74	**18790.76**	19921.30	822.63	**71.94%**	74.46%	0.03
LRC108	**179**	186.62	6.45	19465.68	19750.56	884.63	**75.68%**	77.47%	0.01	24.76	24.82	0.75	197	201.64	5.25	**18770.84**	18745.32	546.80	72.33%	72.75%	0.02

TABLE 7. Experimental results with 300 requests after 10 runs (Multi-mark split has advantage in running time, unit split has advantages in transport vehicle usage (40 out of 60), total distance (46 out of 60), and loading rate (45 out of 60)).

	Unit split												Muti-mark split								
Instance	VN			TD			LR			RT			VN			TD			LR		
	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*	Best	Avg	*std.Dev*
LC101	155	161.21	7.24	**13801.08**	13888.45	375.20	**77.12%**	80.67%	0.03	6.80	6.96	0.06	**132**	135.98	5.12	14189.81	14164.82	126.03	66.69%	65.28%	0.02
LC102	**143**	149.13	7.15	11849.19	12426.73	355.18	**72.12%**	72.04%	0.02	6.64	6.83	0.37	157	162.36	4.73	**11312.58**	11482.14	194.70	66.74%	66.66%	0.02
LC103	**157**	160.63	7.34	**13953.22**	14053.01	658.73	**79.68%**	82.25%	0.01	6.49	6.65	0.24	165	164.53	3.10	14931.53	15686.01	310.31	67.73%	68.52%	0.03
LC104	**141**	149.87	3.09	12133.64	12349.84	643.52	**75.69%**	76.84%	0.03	28.14	28.93	1.56	156	162.89	4.15	**11900.27**	12046.63	479.48	72.35%	73.84%	0.02
LC105	**150**	150.75	8.29	**12947.89**	13374.01	412.03	**81.31%**	82.90%	0.01	3.66	3.73	0.15	159	162.98	4.59	14059.31	14171.87	498.95	75.83%	74.97%	0.03
LC106	**144**	141.15	7.72	**11528.85**	11526.71	409.87	81.30%	83.64%	0.03	2.68	2.71	0.08	152	159.70	2.87	12601.77	12554.29	500.26	**82.80%**	84.70%	0.04
LC107	**162**	166.16	5.70	**15421.47**	15787.40	475.36	**83.26%**	85.04%	0.03	8.50	8.66	0.16	168	170.79	6.41	18571.81	18655.59	755.55	77.69%	79.18%	0.02
LC108	158	160.03	5.02	**14912.18**	14945.51	698.09	**82.48%**	84.32%	0.02	8.59	8.77	0.35	**141**	146.28	6.01	15003.60	14832.74	501.37	68.67%	69.46%	0.02
LC109	**169**	174.45	4.89	**17061.02**	17462.29	583.13	**77.49%**	79.90%	0.03	13.72	13.86	0.60	190	192.86	3.63	18132.41	18774.86	590.27	76.03%	76.70%	0.02
LR101	**193**	195.70	2.47	**15073.42**	15259.81	653.86	**78.82%**	79.34%	0.04	4.44	4.64	0.16	202	208.07	5.67	16064.63	16350.27	309.11	71.61%	71.51%	0.02
LR102	189	190.59	5.85	14863.78	15235.53	312.81	**72.11%**	74.16%	0.03	7.28	7.50	0.30	**182**	186.36	4.52	**13929.55**	14401.56	381.63	66.35%	67.12%	0.01
LR103	**177**	190.24	3.75	**15238.33**	15655.79	339.48	80.65%	81.50%	0.03	14.62	14.86	0.60	193	197.74	4.69	16260.26	15927.74	494.64	**83.53%**	85.97%	0.03
LR104	**184**	191.16	4.14	**15394.59**	15723.35	853.67	**71.53%**	72.26%	0.05	28.76	29.04	1.23	191	197.67	5.22	16468.57	16745.76	347.53	68.78%	71.73%	0.01
LR105	**189**	194.65	6.37	**16637.72**	16653.14	649.23	72.75%	73.88%	0.04	11.09	11.27	0.62	202	207.15	3.41	18743.60	18908.66	897.47	**81.18%**	82.11%	0.02
LR106	174	178.75	3.62	**14301.58**	14406.05	340.80	79.37%	77.76%	0.01	11.12	11.51	0.76	**171**	172.85	3.98	15203.39	15736.17	299.61	**80.26%**	80.93%	0.02
LR107	**160**	160.04	4.28	**14914.45**	15389.01	939.75	**74.34%**	77.15%	0.02	24.17	24.69	0.97	167	172.45	5.37	16656.04	17233.94	339.31	65.70%	67.40%	0.02
LR108	**176**	180.38	6.25	**15760.03**	16407.15	437.56	**77.00%**	78.45%	0.04	19.91	20.31	1.00	201	208.04	4.20	15891.07	15842.32	572.51	65.21%	66.02%	0.03
LR109	**191**	193.58	6.53	**16605.60**	17543.82	606.01	**80.06%**	82.82%	0.04	15.65	15.58	0.45	194	197.73	7.13	20753.00	21163.01	875.76	76.92%	77.30%	0.03
LR110	**171**	174.19	9.55	**15022.01**	15600.70	399.05	81.91%	80.78%	0.02	15.83	16.15	0.54	186	192.72	4.77	15043.40	15283.58	524.66	**83.69%**	85.29%	0.02
LR111	**151**	158.59	4.87	13438.41	13494.72	631.47	**83.77%**	84.85%	0.03	16.01	16.69	0.87	167	173.35	8.36	**12062.45**	12350.43	544.67	79.92%	82.14%	0.02
LR112	**171**	168.87	5.29	**16343.02**	16140.04	454.98	83.82%	85.34%	0.05	12.99	13.35	0.39	191	197.91	4.59	17305.09	17658.86	524.86	**89.49%**	89.13%	0.05
LRC101	**190**	190.81	8.66	**19247.64**	19142.30	584.24	83.88%	87.34%	0.02	9.96	10.05	0.36	229	238.45	8.50	20218.89	20971.01	417.59	**88.44%**	86.96%	0.02
LRC102	192	200.21	8.39	**20147.11**	20138.54	511.01	78.49%	79.27%	0.03	10.71	11.03	0.34	**179**	185.80	4.55	20256.95	20704.96	522.59	**83.79%**	83.90%	0.02
LRC103	**180**	183.96	6.28	**18307.54**	18850.15	391.08	73.18%	75.60%	0.02	22.30	22.37	0.72	211	215.89	8.04	20866.96	21463.49	792.71	**78.54%**	80.84%	0.02
LRC104	**200**	202.50	10.01	19367.79	20587.11	805.69	77.73%	81.19%	0.03	17.17	17.52	0.41	225	237.18	8.88	**17999.88**	18515.46	664.45	**83.68%**	82.80%	0.04
LRC105	**192**	195.83	6.11	**20428.05**	20411.01	617.73	**82.29%**	85.36%	0.02	12.05	12.35	0.60	196	200.67	7.84	21262.70	21202.11	670.80	81.48%	83.07%	0.02
LRC106	**184**	186.91	8.07	**21590.39**	22194.82	698.70	**81.71%**	86.46%	0.02	23.83	24.91	0.66	210	208.67	3.65	23434.82	24550.85	563.01	81.31%	83.26%	0.02
LRC107	193	195.55	7.96	20257.52	21040.69	1071.94	70.29%	70.38%	0.02	28.31	27.96	1.48	**182**	185.96	5.74	**18790.76**	19921.30	822.63	**71.94%**	74.46%	0.03
LRC108	**179**	186.62	6.45	19465.68	19750.56	884.63	**75.68%**	77.47%	0.01	24.76	24.82	0.75	197	201.64	5.25	**18770.84**	18745.32	546.80	72.33%	72.75%	0.02

The experimental results on instances with 50 requests and 300 requests are shown in Table 6 and Table 7, where *RT* represents the algorithm running times in second, *VN* represents the quantity of vehicle used, *LR* represents the percentage of loading rate, and *TD* represents total distance. After 10 runs of each group of data, its best results, average results and standard deviation were analyzed. As shown in Table 6, the unit split has advantages in total distance, transport vehicle usage, and loading rate. Among them, 78.57% (44 out of 56) of the data shows an advantage in transport vehicle usage, and 76.79% (43 out of 56) of the data shows an advantage in total distance, and 60.71% (34 out of 56) of the data shows an advantage in loading rate. As shown in Table 7, the unit split has advantages in total distance, transport vehicle usage, and loading rate. Among them, 66.67% (40 out of 60) of the data shows an advantage in transport vehicle usage,

TABLE 8. Comparison of unit split and multi-mark split algorithms (VN% = $VN_{unit} - VN_{multi-mark} / VN_{unit}$, *TD% and LR% are calculated in the same way*, RT% = $RT_{unit} / RT_{multi-mark}$).

Instance	VN%	TD%	LR%	RT%	Instance	VN%	TD%	LR%	RT%
	Deviation					Deviation			
LC101	0.15	-0.03	0.14	5.80	LC1_6_1	-0.12	-0.16	0.00	5.42
LC102	-0.10	0.05	0.07	4.53	LC1_6_2	-0.06	-0.07	0.05	5.72
LC103	-0.05	-0.07	0.15	4.52	LC1_6_3	0.05	-0.10	0.13	4.62
LC104	-0.11	0.02	0.04	5.73	LC1_6_4	-0.04	-0.03	0.06	5.17
LC105	-0.06	-0.09	0.07	4.44	LC1_6_5	-0.14	-0.13	-0.03	4.67
LC106	-0.06	-0.09	-0.02	5.40	LC1_6_6	0.13	-0.05	0.04	4.75
LC107	-0.04	-0.20	0.07	5.30	LC1_6_7	-0.18	-0.03	0.14	5.08
LC108	0.10	-0.01	0.17	5.74	LC1_6_8	-0.01	-0.06	0.01	4.71
LC109	-0.12	-0.06	0.02	4.75	LC1_6_9	0.06	0.03	-0.05	5.25
LR101	-0.05	-0.07	0.09	6.11	LC1_6_10	0.01	-0.11	0.02	4.93
LR102	0.03	0.06	0.08	5.12	LR1_6_1	0.05	0.13	0.03	5.15
LR103	-0.09	-0.07	-0.04	4.75	LR1_6_2	-0.02	-0.16	0.06	4.65
LR104	-0.04	-0.07	0.04	4.85	LR1_6_3	-0.01	-0.12	0.01	5.04
LR105	-0.07	-0.13	-0.12	5.09	LR1_6_4	0.02	0.01	-0.10	4.73
LR106	0.01	-0.06	-0.01	5.54	LR1_6_5	-0.01	-0.04	0.05	5.51
LR107	-0.05	-0.12	0.12	4.86	LR1_6_6	-0.10	0.04	-0.06	4.62
LR108	-0.15	-0.01	0.15	5.62	LR1_6_7	0.04	-0.14	0.00	5.07
LR109	-0.01	-0.25	0.04	4.65	LR1_6_8	0.02	-0.07	0.20	4.54
LR110	-0.09	0.00	-0.02	5.18	LR1_6_9	-0.05	-0.08	0.02	4.93
LR111	-0.10	0.10	0.05	5.65	LR1_6_10	-0.14	-0.05	0.13	4.37
LR112	-0.12	-0.06	-0.07	4.53	LRC1_6_1	0.03	-0.08	0.06	5.90
LRC101	-0.20	-0.05	-0.05	4.63	LRC1_6_2	-0.10	-0.12	0.09	5.06
LRC102	0.07	-0.01	-0.07	4.69	LRC1_6_3	-0.09	-0.03	-0.03	5.08
LRC103	-0.17	-0.14	-0.07	5.41	LRC1_6_4	0.02	0.04	0.18	5.26
LRC104	-0.13	0.07	-0.08	4.82	LRC1_6_5	-0.06	0.05	-0.09	5.41
LRC105	-0.02	-0.04	0.01	4.78	LRC1_6_6	-0.16	-0.02	0.14	4.23
LRC106	-0.14	-0.09	0.00	4.62	LRC1_6_7	0.10	-0.24	0.04	5.32
LRC107	0.06	0.07	-0.02	5.13	LRC1_6_8	-0.07	-0.03	0.03	5.30
LRC108	-0.10	0.04	0.04	4.79	LRC1_6_9	-0.14	0.05	-0.10	5.35
					LRC1_6_10	-0.20	0.11	0.14	4.69

TABLE 9. Parameters and cost information of transportation vehicles.

TVT	RC	E	F	velocity（km/h）
L1	6	198	0.853	60
L11	8	223	1.214	60
L12	10	544	2.126	60
L22	12	885	3.682	60

TVT represents transport vehicle type, RC represents the rated capacity of transport vehicle (unit: ton), E represents fixed cost (unit: ￥/times), F represents the unit loading cost (unit: ￥/ vehicle · km).

and 76.67% (46 out of 60) of the data shows an advantage in total distance, and 75% (45 out of 60) of the data shows an advantage in loading rate. But the multi-mark split algorithm has an obvious advantage in running time (about 5 times less on average). At the same time, the multi-mark split algorithm has a slight advantage in algorithm stability.

Table 8 reports a more accurate comparison of the deviation values of the two algorithms in multiple dimensions. As shown in the table, the deviation between the unit split method and multi-mark split method in the *NV*, *TD*, and *LR* is very small. The average deviation of the vehicle usage is 8.47%, the average deviation of distance is 6.87%, and the average deviation of the loading rate is 5.75%. Therefore,

TABLE 10. **The unit loading cost of transport vehicles (unit: ￥/*vehicle* · *km*).**

PVT \ TVT	L1	L11	L12	L22
MPV	2.25	2	1.5	1
SUV	2	1.78	1.33	0.88
compact cars	1.8	1.6	1.2	0.8

PVT represents passenger finished vehicle type

TABLE 11. **The structure of partial orders (other represents compact cars).**

Order number	Demands	Retention time /day	Pickup address Name	Pickup address Latitude	Pickup address Longitude	Delivery address Name	Delivery address Latitude	Delivery address Longitude	Amount / ￥	Vehicle type	Latest delivery time/day
1	6	12	Shunyi	40.15495	116.7282	Wuhai	39.68318	106.832	846	other	30
2	7	17	Tianjin	40.15495	116.7282	Zhangjiakou	40.81119	114.8938	342	other	30
3	7	15	Shunyi	40.15495	116.7282	Liupanshui	26.59187	104.8521	3732	SUV	30
4	6	15	Tianjin	40.15495	116.7282	Baoding	38.88657	115.4948	176	SUV	30
5	9	14	Cangzhou	40.15495	116.7282	Shanwei	22.77873	115.3729	7752	SUV	30
6	7	12	Shunyi	40.15495	116.7282	Yucheng	36.91914	116.5813	720	other	30
7	6	21	Shunyi	40.15495	116.7282	Taiyuan	37.89028	112.5509	440	SUV	30
8	10	11	Shunyi	40.15495	116.7282	Chengde	40.99252	117.9338	690	SUV	30
…	…	…	…	…	…	…	…	…	…	…	…
200	6	16	Wuhan	30.58108	114.3162	Guangde	30.89395	119.3647	484	other	30

we strongly suggest to use the multi-mark split algorithm in scenes with high real-time requirements or large-scale datasets, while to use the unit split algorithm in scenes with high accuracy requirements or small-scale datasets.

B. VALIDATION ON THE ACTUAL INSTANCE

In order to verify the practical value of the model and algorithm, the verification is also carried out on the data of an actual enterprise. Company C has four types of transport vehicles of single-layer L1, double-layer L11, L12, and L22. Calculate the purchase costs, insurance costs, labor costs, fuel consumption, road and bridge costs, maintenance costs and tire loss costs of these transport vehicles, and calculate the fixed and variable costs for each transport vehicle. The parameters and cost information of transport vehicles are shown in Table 9, where TVT represents the transport vehicle type, RC represents the rated capacity of transport vehicle(unit: ton), E represents the fixed cost(unit: ￥/times), F represents the unit loading cost (unit: ￥/ $vehicle \cdot km$). The unit loading cost of the transport vehicle is a variable cost of the transport vehicle transport one passenger finished vehicle for one kilometer. The passenger finished vehicles studied in this article mainly include SUV, MPV, and compact cars. The unit loading costs of transport vehicles are shown in Table 10, where PVT represents the passenger finished vehicle type.

We select the partial data with 200 orders and 500 transport vehicles of company C on a day to verify the algorithm.

TABLE 12. **Experimental results on actual instance.**

	Total profit /￥	RT / seconds	VN	LR
Our algorithm	322577.93	197.28	138	82.36
Manual operation	268513.87	-	186	62.85
Deviation	20.13%	-	-25.81%	31.04%

The data of orders and transport vehicles are given in attachments task.xlsx and vehicle.xlsx. Table 11 shows the structure of partial orders.

The length of the tabu list of the algorithm in this article is set to 1000. After running 10 times, we take the data with the largest profit as a result. As shown in Table 12, comparing the results of our algorithm with the results of enterprise statistics, this algorithm has obvious advantages in terms of total profit, number of vehicles and loading rate. Our algorithm can complete in an average of about 3 minutes, and the average loading rate of transport vehicles can reach more than 80%. Therefore, the model and algorithm in this article have certain use-value, and can effectively reduce the transportation cost of enterprises.

VI. CONCLUSION

This article has studied the vehicle scheduling problem of third-party passenger finished vehicle logistics, and an integer programming model is established to maximize the total profit. As far as we know, the path buffer clustering operator is the first time proposed to be applied to vehicle routing planning problems, and is more effective compared to the k-means clustering, and is more practical in actual engineering application. The multi-mark split operator can save a lot of running time at the expense of little algorithm precision, and is strongly suggested to be used in scenes with high real-time requirements or large-scale datasets. The experimental results show that the algorithm proposed in this article is efficient and effective, and can effectively improve the average loading rate of vehicles to about 80% in a relatively short time.

Our sensitive experiments on α, β, γ, and δ show that the values of these four key parameters of PCB are closely related to the geographical distribution of orders and transport vehicles, in actual application, in order to get the ideal optimum objective value, these parameters are suggested to be set as the empirical formulas provided by this article. While sensitive experiments on the split proportion which is a key parameter of MMS show that with the increase of split proportion, the loading rate of transport vehicles tends to increase, the number of transport vehicles used tends to decrease, and that with the increase of split proportion,the max profit of the enterprise increases firstly and then decrease quickly, the maximum profit can be obtained at the point of that the value of split proportion is about 85%.

Although the algorithm has achieved some excellent results, the following is worthy of further study:

1) The actual algorithm complexity of path buffer clustering combining with an actual geographic information system.
2) The dynamic priority to the requests, that is, to get the maximum profit not all the requests in a scheduling must be served.
3) Multiple continuous scheduling to maximize the total profit.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and all the reviewers for their constructive comments and suggestions, which improve the quality of this article.

REFERENCES

[1] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis, "Vehicle routing strategies for pick-up and delivery service under two dimensional loading constraints," *Oper. Res.*, vol. 17, no. 1, pp. 115–143, Apr. 2017.

[2] Z. Li and Y. Zhang, "Research on loading and routing problem of finished car logistic," in *Proc. Int. Conf. Logistics, Inform. Service Sci. (LISS)*, 2016, pp. 1–6.

[3] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants," *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106242.

[4] F. Ferrucci, S. Bock, and M. Gendreau, "A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 130–141, Feb. 2013.

[5] X. Zhang, Y. Chen, and Q. Wang, "Planning model and algorithm for finished vehicle logistics," *China Manage. Sci.*, vol. S1, pp. 624–629, 2015.

[6] Z.-H. Hu, Y. Zhao, S. Tao, and Z.-H. Sheng, "Finished-vehicle transporter routing problem solved by loading pattern discovery," *Ann. Oper. Res.*, vol. 234, no. 1, pp. 37–56, Nov. 2015.

[7] Y.-N. Jiang, Q. Xu, H. Ren, and Z.-H. Jin, "Vehicle logistics path optimization under resource sharing mode," *Highway Transp. Technol.*, vol. 34, no. 6, pp. 114–121, 2017.

[8] M. Dror and P. Trudeau, "Savings by split delivery routing," *Transp. Sci.*, vol. 23, no. 2, pp. 141–145, May 1989.

[9] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza, "Worst-case analysis for split delivery vehicle routing problems," *Transp. Sci.*, vol. 40, no. 2, pp. 226–234, May 2006.

[10] C. Archetti, N. Bianchessi, and M. G. Speranza, "A column generation approach for the split delivery vehicle routing problem," *Networks*, vol. 58, no. 4, pp. 241–254, Dec. 2011.

[11] G. Desaulniers, "Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows," *Oper. Res.*, vol. 58, no. 1, pp. 179–192, Feb. 2010.

[12] C. Archetti, D. Feillet, M. Gendreau, and M. G. Speranza, "Complexity of the VRP and SDVRP," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 5, pp. 741–750, Aug. 2011.

[13] C. Archetti and M. G. Speranza, "Vehicle routing problems with split deliveries," *Int. Trans. Oper. Res.*, vol. 19, nos. 1–2, pp. 3–22, Jan. 2012.

[14] J. M. Belenguer, M. C. Martinez, and E. Mota, "A lower bound for the split delivery vehicle routing problem," *Oper. Res.*, vol. 48, no. 5, pp. 801–810, Oct. 2000.

[15] C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza, "To split or not to split: That is the question," *Transp. Res. E, Logistics Transp. Rev.*, vol. 44, no. 1, pp. 114–123, 2008.

[16] M. Jin, K. Liu, and R. O. Bowden, "A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem," *Int. J. Prod. Econ.*, vol. 105, no. 1, pp. 228–242, Jan. 2007.

[17] L. Moreno, M. P. de Aragão, and E. Uchoa, "Improved lower bounds for the split delivery vehicle routing problem," *Oper. Res. Lett.*, vol. 38, no. 4, pp. 302–306, Jul. 2010.

[18] C. Archetti, N. Bianchessi, and M. G. Speranza, "A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem," *Comput. Oper. Res.*, vol. 64, pp. 1–10, Dec. 2015.

[19] V. Campos, A. Corberán, and E. Mota, "A scatter search algorithm for the split delivery vehicle routing problem," in *Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management*. Berlin, Germany: Springer, 2008, pp. 137–152.

[20] S. Yan, J. C. Chu, F.-Y. Hsiao, and H.-J. Huang, "A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows," *Comput. Ind. Eng.*, vol. 87, pp. 383–393, Sep. 2015.

[21] S. Chen, B. Golden, and E. Wasil, "The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results," *Networks*, vol. 49, no. 4, pp. 318–329, Jul. 2007.

[22] C. Archetti, M. G. Speranza, and M. W. P. Savelsbergh, "An optimization-based heuristic for the split delivery vehicle routing problem," *Transp. Sci.*, vol. 42, no. 1, pp. 22–31, Feb. 2008.

[23] C. Archetti, M. G. Speranza, and A. Hertz, "A tabu search algorithm for the split delivery vehicle routing problem," *Transp. Sci.*, vol. 40, no. 1, pp. 64–73, Feb. 2006.

[24] M. Jin, K. Liu, and B. Eksioglu, "A column generation approach for the split delivery vehicle routing problem," *Oper. Res. Lett.*, vol. 36, no. 2, pp. 265–270, Mar. 2008.

[25] A. Khmelev and Y. Kochetov, "A hybrid VND method for the split delivery vehicle routing problem," *Electron. Notes Discrete Math.*, vol. 47, pp. 5–12, Feb. 2015.

[26] U. Derigs, B. Li, and U. Vogel, "Local search-based metaheuristics for the split delivery vehicle routing problem," *J. Oper. Res. Soc.*, vol. 61, no. 9, pp. 1356–1364, Sep. 2010.

[27] R. E. Aleman, X. Zhang, and R. R. Hill, "An adaptive memory algorithm for the split delivery vehicle routing problem," *J. Heuristics*, vol. 16, no. 3, pp. 441–473, Jun. 2010.

[28] R. E. Aleman and R. R. Hill, "A tabu search with vocabulary building approach for the vehicle routing problem with split demands," *Int. J. Metaheuristics*, vol. 1, no. 1, pp. 55–80, 2010.

[29] J. H. W. Iv and T. M. Cavalier, "A genetic algorithm for the split delivery vehicle routing problem," *Amer. J. Oper. Res.*, vol. 2, no. 2, pp. 207–216, 2012.

[30] L. Berbotto, S. García, and F. J. Nogales, "A randomized granular tabu search heuristic for the split delivery vehicle routing problem," *Ann. Oper. Res.*, vol. 222, no. 1, pp. 153–173, Nov. 2014.

[31] M. M. Silva, A. Subramanian, and L. S. Ochi, ''An iterated local search heuristic for the split delivery vehicle routing problem,'' *Comput. Oper. Res.*, vol. 53, pp. 234–249, Jan. 2015.

[32] X. Hao and Y. HuiLi, ''A three-phase tabu search heuristic for the split delivery vehicle routing problem,'' *Syst. Eng. Theory Pract.*, vol. 35, no. 5, pp. 1230–1235, 2015.

[33] H. Li and A. Lim, ''A Metaheuristic for the pickup and delivery problem with time windows,'' *Int. J. Artif. Intell. Tools*, vol. 12, no. 02, pp. 173–186, Jun. 2003.

[34] M. Salani and I. Vacca, ''Branch and price for the vehicle routing problem with discrete split deliveries and time windows,'' *Eur. J. Oper. Res.*, vol. 213, no. 3, pp. 470–477, Sep. 2011.

[35] Z. Luo, H. Qin, W. Zhu, and A. Lim, ''Branch and price and cut for the split-delivery vehicle routing problem with time windows and linear weight-related cost,'' *Transp. Sci.*, vol. 51, no. 2, pp. 668–687, May 2017, doi: 10.1287/trsc.2015.0666.

[36] A. Bortfeldt and J. Yi, ''The split delivery vehicle routing problem with three-dimensional loading constraints,'' *Eur. J. Oper. Res.*, vol. 282, no. 2, pp. 545–558, Apr. 2020, doi: 10.1016/j.ejor.2019.09.024.

[37] T. Gschwind, N. Bianchessi, and S. Irnich, ''Stabilized branch-price-and-cut for the commodity-constrained split delivery vehicle routing problem,'' *Eur. J. Oper. Res.*, vol. 278, no. 1, pp. 91–104, Oct. 2019, doi: 10.1016/j.ejor.2019.04.008.

[38] W. Gu, D. Cattaruzza, M. Ogier, and F. Semet, ''Adaptive large neighborhood search for the commodity constrained split delivery VRP,'' *Comput. Oper. Res.*, vol. 112, Dec. 2019, Art. no. 104761.

[39] J. Shi, J. Zhang, K. Wang, and X. Fang, ''Particle swarm optimization for split delivery vehicle routing problem,'' *Asia–Pacific J. Oper. Res.*, vol. 35, no. 2, Apr. 2018, Art. no. 1840006, doi: 10.1142/s0217595918400067.

[40] M. N. Haddad, R. Martinelli, T. Vidal, S. Martins, L. S. Ochi, M. J. F. Souza, and R. Hartl, ''Large neighborhood-based metaheuristic and branch-and-price for the pickup and delivery problem with split loads,'' *Eur. J. Oper. Res.*, vol. 270, no. 3, pp. 1014–1027, Nov. 2018, doi: 10.1016/j.ejor.2018.04.017.

[41] M. Qiu, Z. Fu, R. Eglese, and Q. Tang, ''A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups,'' *Comput. Oper. Res.*, vol. 100, pp. 102–116, Dec. 2018, doi: 10.1016/j.cor.2018.07.021.

[42] M. E. McNabb, J. D. Weir, R. R. Hill, and S. N. Hall, ''Testing local search move operators on the vehicle routing problem with split deliveries and time windows,'' *Comput. Oper. Res.*, vol. 56, pp. 93–109, Apr. 2015.

[43] Q. Chen, K. Li, and Z. Liu, ''Model and algorithm for an unpaired pickup and delivery vehicle routing problem with split loads,'' *Transp. Res. E, Logistics Transp. Rev.*, vol. 69, pp. 218–235, Sep. 2014.

[44] Y. Wang, X.-L. Ma, Y.-T. Lao, H.-Y. Yu, and Y. Liu, ''A two-stage heuristic method for vehicle routing problem with split deliveries and pickups,'' *J. Zhejiang Univ. Sci. C*, vol. 15, no. 3, pp. 200–210, Mar. 2014.

[45] K. Wang, C. Ye, and A. Ning, ''Achieving better solutions for vehicle routing problem involving split deliveries and pickups using a competitive decision algorithm,'' *Asia–Pacific J. Oper. Res.*, vol. 32, no. 4, Aug. 2015, Art. no. 1550022.

[46] J.-J. Salazar-González and B. Santos-Hernández, ''The split-demand one-commodity pickup-and-delivery travelling salesman problem,'' *Transp. Res. B, Methodol.*, vol. 75, pp. 58–73, May 2015.

[47] P. Belfiore and H. T. Y. Yoshizaki, ''Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries,'' *Comput. Ind. Eng.*, vol. 64, no. 2, pp. 589–601, Feb. 2013.

[48] Y. Kergosien, C. Lenté, J.-C. Billaut, and S. Perrin, ''Metaheuristic algorithms for solving two interconnected vehicle routing problems in a hospital complex,'' *Comput. Oper. Res.*, vol. 40, no. 10, pp. 2508–2518, Oct. 2013.

[49] A. Hertz, M. Uldry, and M. Widmer, ''Integer linear programming models for a cement delivery problem,'' *Eur. J. Oper. Res.*, vol. 222, no. 3, pp. 623–631, Nov. 2012.

[50] J. Lee and B.-I. Kim, ''Industrial ship routing problem with split delivery and two types of vessels,'' *Expert Syst. Appl.*, vol. 42, no. 22, pp. 9012–9023, Dec. 2015.

[51] J. N. Min, C. Jin, and L. J. Lu, ''Maximum-minimum distance clustering method for split-delivery vehicle-routing problem: Case studies and performance comparisons,'' *Adv. Prod. Eng. Manage.*, vol. 14, no. 1, pp. 125–135, Mar. 2019, doi: 10.14743/apem2019.1.316.

[52] Y. Xia and Z. Fu, ''An adaptive tabu search algorithm for the open vehicle routing problem with split deliveries by order,'' *Wireless Pers. Commun.*, vol. 103, no. 1, pp. 595–609, Nov. 2018, doi: 10.1007/s11277-018-5464-4.

[53] Y. R. Shi, D. F. Wan, S. Y. Li, and Z. Y. Lv, ''Research of vehicle routing problem based on GIS,'' *Syst. Eng. Theory Pract.*, vol. 29, no. 10, pp. 76–84, 2009.

[54] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, ''Heuristics for multi-attribute vehicle routing problems: A survey and synthesis,'' *Eur. J. Oper. Res.*, vol. 231, no. 1, pp. 1–21, Nov. 2013.

[55] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, ''Understanding of internal clustering validation measures,'' in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 911–916.

[56] N. Wang, M. Zhang, A. Che, and B. Jiang, ''Bi-objective vehicle routing for hazardous materials transportation with no vehicles travelling in echelon,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1867–1879, Jun. 2018, doi: 10.1109/TITS.2017.2742600.

[57] M. Zhang, N. Wang, Z. He, Z. Yang, and Y. Guan, ''Bi-objective vehicle routing for hazardous materials transportation with actual load dependent risks and considering the risk of each vehicle,'' *IEEE Trans. Eng. Manag.*, vol. 66, no. 3, pp. 429–442, Aug. 2019, doi: 10.1109/TEM.2018.2832049.

ZHIYUAN SU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2005. He is currently teaching at the Beijing University of Posts and Telecommunications. His research interests include logistics equipment and automation, logistics systems, computer control, logistics information systems, and software system architecture.

WANTAO LI is currently a Graduate Student with the Beijing University of Posts and Telecommunications. He is also working as an Intern at Meituan to study the algorithm and strategy of takeaway distribution. His research interest includes logistics informatization.